

Entwicklung von computergestützten Lehr- und Lernmaterialien für einen realitätsbezogenen Mathematikunterricht zum Kontext Sprachverarbeitung

Zur Erlangung des akademischen Grades einer

DOKTORIN DER NATURWISSENSCHAFTEN

von der KIT-Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von

Stephanie Hofmann

Tag der mündlichen Prüfung: 23.04.2025

Erstgutachter: Prof. Dr. Martin Frank

Zweitgutachter: Prof. i. R. Dr. Norbert Henze

Drittgutachter: Prof. Dr. Sebastian Bauer



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung -
Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-SA 4.0):
<https://creativecommons.org/licenses/by-sa/4.0/deed.de>

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

.....
Stephanie Hofmann
Karlsruhe, 25.02.2025

Danksagung

Eine Promotion ist wie ein Weg durch eine Gebirgslandschaft. Es gibt Zeiten des Hochgefühls, ähnlich dem Gefühl, einen Gipfel erklimmen zu haben: Erfolge bei Materialtests, die Annahme eines Artikels, positives Feedback von Lehrkräften zum Lernmaterial oder natürlich die erfolgreiche Einreichung der Dissertation. Aber es gibt auch Zeiten, wenn etwas nicht nach Plan läuft, in denen man sich dann fühlt, als müsse man einen steinigen Abgrund überwinden. Allen Personen, die mich in Höhen und Tiefen auf diesem Weg begleitet und unterstützt haben und mir somit ermöglicht haben, den Gipfel zu erreichen, möchte ich meinen Dank aussprechen.

Ich danke meinem Betreuer *Prof. Dr. Martin Frank*, der mir diese Promotion ermöglicht hat und mir sehr viel Raum gegeben hat, diejenigen Themen zu erforschen, die auch ich interessant fand. Er hat mich vor allem durch Denkanstöße und konstruktives Feedback, aber auch durch seine Ermutigungen und dem Vertrauen, was er mir und meinen Fähigkeiten entgegengebracht hat, sehr unterstützt.

Meinem Zweitbetreuer *Prof. i. R. Dr. Norbert Henze* möchte ich ebenfalls meinen Dank aussprechen. Er hat mir mit seinem umfassenden Fachwissen in der Stochastik und seiner sprachlichen Expertise sehr geholfen und insbesondere in der finalen Phase wesentlich dazu beigetragen, diese Dissertation zu verbessern.

Auch bei *Prof. Dr. Sebastian Bauer* möchte ich mich sehr für sein wertvolles Feedback und die hilfreichen Gespräche zur stoffdidaktischen Rahmung dieser Arbeit bedanken.

Mein Dank gilt zudem dem Projekt Simulierte Welten, über das diese Promotion finanziert wurde. Die Arbeit im zugehörigen Schülerprojekt CAMMP war für mich eine bereichernde Erfahrung, und ich bin dankbar, Teil eines so inspirierenden Projekts mit einer großartigen Philosophie und einem hervorragenden Arbeitsklima gewesen zu sein. Besonderer Dank gebührt meinen aktuellen und ehemaligen Kolleginnen und Kollegen des CAMMP-Teams – *Jakim Eckert, Dr. Sarah Schönbrodt, Stephan Kindler, Dr. Katharina Bata* und *Dr. Kirsten Wohak* – für den wertvollen fachlichen Austausch zu meinem Lernmaterial, ihr konstruktives Feedback zu einzelnen Kapiteln dieser Dissertation und die äußerst motivierte und respektvolle Zusammenarbeit. Sie sind in dieser Zeit zu wahren Freunden geworden.

Außerdem möchte ich mich bei meinen Kollegen *Jaroslav Borodavka* und *Dr. Leonid Chaichenets* und ganz besonders bei meiner ehemaligen Kollegin *Dr. Lea Schenk* für hilfreiche Anmerkungen, fachliche Diskussionen und aufmunternde Worte bedanken.

Zuletzt gebührt mein Dank meiner *Familie* und meinen *Freundinnen und Freunden*, die unabhängig von den Entscheidungen, die ich getroffen habe, mit ihrer Überzeugung, dass ich das alles schaffen kann, den Rücken gestärkt haben. Mein größter

Dank gilt dabei meinem Freund *Josua Marlok*, der mich insbesondere in der letzten Phase dieser Arbeit emotional unterstützt und mir den nötigen Freiraum geschaffen hat, mich ganz auf die Dissertation zu konzentrieren.

Vorwort

Die kontinuierliche Entwicklung im Bereich der Künstlichen Intelligenz (KI) hat in den letzten Jahren zu erstaunlichen Fortschritten insbesondere in der Verarbeitung natürlicher Sprache (NLP) geführt. Spracherkennung und Chatbots wie ChatGPT sind nur zwei prominente Beispiele dafür. In den letzten Jahren hat vor allem die Informatik bei jungen Erwachsenen an enormer Popularität gewonnen – vermutlich stark durch die Begeisterungswelle um Systeme Künstlicher Intelligenz beeinflusst. Zwischen den Wintersemestern 2013/2014 und 2023/2024 hat sich die Zahl der Erstsemester im Fach Informatik beinahe verdoppelt (Statistisches Bundesamt, [2024](#)). Ein solcher Anstieg ist bei den Studienanfängerinnen und -anfängern des Faches Mathematik nicht zu beobachten. Die Tatsache, dass viele KI- und NLP-Technologien auf komplexen mathematischen Modellen basieren, scheint ein Aspekt zu sein, der in der öffentlichen Wahrnehmung oft nur eine untergeordnete Rolle spielt. Genau hier liegt jedoch eine große Chance für den Mathematikunterricht. Wenn Schülerinnen und Schüler erkennen, dass hinter aktuellen, realen Problemen wie der Spracherkennung in KI-Systemen wesentliche mathematische Konzepte stehen, könnte dies ihr Verständnis für die Bedeutung der Mathematik vertiefen und ihre Motivation für das Fach erhöhen.

Aufgrund der verstärkten Anwendung von KI und NLP im täglichen Leben ist es zudem von entscheidender Bedeutung, dass Lernende ein grundlegendes Verständnis für diese Technologien entwickeln (Long & Magerko, [2020](#), S. 1; Mertala & Fagerlund, [2023](#), S. 12 - 13; Miao & Shiohira, [2024](#), S. 3). Im Mathematikunterricht können notwendige Kenntnisse vermittelt werden, um grundlegende Funktionsweisen dieser Technologien zu verstehen, sie kritisch zu bewerten und reflektiert anwenden zu können, wodurch die Lernenden zu einem mündigen Umgang mit KI-Systemen befähigt werden.

Genau hier setzt diese Arbeit an und widmet sich der Auswahl, Analyse und Aufbereitung mathematischer Inhalte im Bereich der Verarbeitung natürlicher Sprache, einem hochrelevanten Beispiel aus der Praxis und Teilbereich des Forschungsgebietes der Künstlichen Intelligenz. Ziel ist es, aktuelle, relevante, authentische Problemstellungen in den Mathematikunterricht zu integrieren und damit den Schülerinnen und Schülern nicht nur die Relevanz der Mathematik, sondern auch deren Beitrag zur Entwicklung moderner Anwendungen zu verdeutlichen. Gleichzeitig soll über die Funktionsweise von KI-Systemen und deren mathematische Grundlagen aufgeklärt werden, um ein vertieftes Verständnis von KI zu fördern.

Diese Arbeit lässt sich, sofern es einer Einordnung bedarf, dem Bereich der Stoffdidaktik zuordnen, der sich mit der Analyse von Inhalten und Fragestellungen für den Mathematikunterricht beschäftigt. Wie Hefendehl-Hebeker ([2022](#), S. 2) betont,

gehört diese Analyse zu den zentralen Aufgaben der Stoffdidaktik. Die Entwicklung und die Erprobung von Lernmaterial auf der Grundlage theoretischer und praktischer Einsichten mit dem Ziel einer Bereicherung des Unterrichts sieht Hefendehl-Hebeker (2022, S. 12) dafür als geeignete Methode an. Im Fokus steht deshalb die Entwicklung und Erprobung eines Lehr-Lernarrangements für den Mathematikunterricht, welches die Modellierung von Problemstellungen aus dem Bereich KI und insbesondere aus dessen Teilbereich NLP erlaubt.

In der vorliegenden Arbeit werden dazu beispielhaft zwei verschiedene Problemstellungen aus dem Kontext des NLP untersucht, deren Auswahl nach verschiedenen Kriterien erfolgt. Entscheidend ist unter anderem, dass die Problemstellungen aus Anwendungskontexten stammen, die eine hohe Alltagsrelevanz für die Lernenden besitzen und tatsächlich auch Gegenstand der Forschung sind.

Die erste Problemstellung befasst sich mit der Funktionsweise von Assistenzsystemen, die Nutzende beim Schreiben einer Nachricht auf dem Smartphone unterstützen. Konkret wird untersucht, wie Vorschläge für das nächste Wort beim Chatten generiert werden und welche Rolle die Mathematik hierbei spielt. Im Lernmaterial wird die Entwicklung eines statistischen Sprachmodells ins Zentrum gestellt. Zudem wird ein Bezug zum hochaktuellen Thema der generativen KI hergestellt, indem das entwickelte Sprachmodell auch zur Generierung von Texten eingesetzt wird.

Die zweite Problemstellung widmet sich der Erkennung von gesprochener Sprache. Die Motivation für die Wahl dieser Problemstellung liegt unter anderem in der Relevanz von Sprachtechnologie für den Alltag der Lernenden, insbesondere durch die alltägliche Präsenz von Sprachdialogsystemen wie Siri und Alexa. Diese Arbeit soll zeigen, dass die grundlegende Funktionsweise solcher höchst komplex und undurchdringlich wirkenden Systeme durchaus mit Lernenden erarbeitet werden kann.

Aufbau dieser Arbeit

Im Folgenden soll der Aufbau dieser in fünf Teile gegliederten Arbeit dargelegt werden, um eine Orientierung zu erleichtern.

Teil I legt dar, weshalb Modellierungsprobleme aus dem Anwendungskontext der Verarbeitung natürlicher Sprache im Mathematikunterricht behandelt werden sollten. Dieser Teil legitimiert das Entwicklungsziel und dient der Konkretisierung des Ziels dieser Arbeit. Hierzu wird zunächst erläutert, was unter mathematischer Modellierung verstanden wird und wie sie betrieben werden kann, um später als Grundlage für das Lösen von realen Problemen aus dem Bereich der Verarbeitung natürlicher Sprache zu dienen. Dabei wird ein besonderes Augenmerk auf die Mathematische Modellierung unter Einsatz digitaler Medien gelegt. Es wird begründet, weshalb Modellierungsaufgaben in den Mathematikunterricht integriert werden sollten und damit die Relevanz eines stärker realitätsbezogenen Unterrichts untermauert. Zudem wird das Entwicklungsziel legitimiert, indem Gründe für die Behandlung von Themen der Künstlichen Intelligenz, insbesondere der Verarbeitung natürlicher Sprache, im Mathematikunterricht aufgezeigt werden. Darüber hinaus präzisiert dieser Teil

das Forschungsvorhaben, gibt einen Überblick über den aktuellen Forschungsstand der (Mathematik-)Didaktik im Bereich Sprachverarbeitung und Künstliche Intelligenz und grenzt die vorliegende Arbeit von verwandten Forschungsprojekten ab.

In Teil **II** werden übergreifende Lernziele für die Lernmaterialien beider Modellierungsprobleme formuliert. Zudem erfolgt eine Vorstellung und Begründung der Design-Prinzipien, welche als Orientierung für die Entwicklung der Lernmaterialien dienen und die Erreichung der aufgestellten Lernziele unterstützen sollen. Darüber hinaus wird in diesem Teil der ganzheitliche Ansatz zur Durchführung kollaborativer Modellierungsprojekte vorgestellt, der im Rahmen des *Computational and Mathematical Modeling Programs* (CAMMP) entwickelt wurde. Dazu zählt auch eine Beschreibung und Untersuchung des verwendeten digitalen Werkzeugs sowie dessen Ausgestaltung als digitale Lernumgebung und eine Vorstellung der grundsätzlichen Abläufe der Modellierungsprojekte.

Teil **III** und Teil **IV** dieser Arbeit zeigen anhand zweier Beispiele, wie die Thematisierung der Verarbeitung natürlicher Sprache im Mathematikunterricht gelingen kann. Hierzu wird der Hintergrund der beiden genannten Problemstellungen zunächst mathematisch-inhaltlich untersucht und die jeweiligen realen Probleme mathematisch modelliert. Zentrale mathematische Konzepte der Verarbeitung natürlicher Sprache, speziell die der Spracherkennung und Wortvorhersage, werden identifiziert und spezifiziert. Auf Basis dessen werden die mathematischen Methoden didaktisch reduziert, sodass sie für Schülerinnen und Schüler der Jahrgangsstufe 9 beziehungsweise 10 zugänglich sind. Schließlich werden Aufgaben entwickelt und zu einer kohärenten Lehr-Lernumgebung zusammengeführt. Es erfolgt eine curriculare Einordnung der entwickelten Lernmaterialien. Der nächste Schritt stellt schließlich die Beobachtungen bei Erprobungen des Materials dar. Dazu wurde das Material im Rahmen verschiedener Modellierungsveranstaltungen des Programmes CAMMP eingesetzt und evaluiert. Die dabei gemachten Erfahrungen mit den entwickelten Lernmaterialien und daraus resultierende Anpassungen der Materialien werden hier dargelegt. Abschließend werden weiterführende Forschungsaspekte zu den spezifischen Materialien beleuchtet.

Teil **V** fasst die Arbeit knapp zusammen und gibt einen allgemeinen Ausblick auf anknüpfende Forschungsmöglichkeiten.

Die vorliegende Arbeit baut auf Inhalten auf, die die Autorin bereits in verschiedenen Publikationen vorgestellt hat und ergänzt diese. Dazu zählen Veröffentlichungen zum Lernmaterial zur Wortvorhersage im Tagesformat (Hofmann & Frank, 2022; Hofmann & Frank, 2023) sowie im Doppelstundenformat (Hofmann & Frank, 2024). Die übergeordneten Lernziele für KI-Modellierungsprojekte, die im Rahmen dieser Arbeit vorgestellt werden, wurden von der Autorin erarbeitet und sind in Geisler et al. (*under review*) beschrieben. Darüber hinaus wurden die Workshops als Bestandteil eines mehrtägigen außerunterrichtlichen Lernangebots für junge Frauen zum Thema KI integriert, dessen Konzept in Hofmann und Schönbrodt (2024) und Hofmann et al. (2024) thematisiert wird.

Inhaltsverzeichnis

Vorwort	v
Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xv
I Theoretischer Rahmen	1
1 Einleitung	3
2 Mathematische Modellierung	5
2.1 Grundlagen der Mathematischen Modellierung	5
2.2 Mathematische Modellierung in der Schule	7
2.3 Mathematische Modellierung und das Zusammenspiel mit digitalen Medien	9
3 Verarbeitung natürlicher Sprache und Künstliche Intelligenz	13
3.1 Verarbeitung natürlicher Sprache	13
3.2 Künstliche Intelligenz und Maschinelles Lernen	14
3.3 Künstliche Intelligenz und Verarbeitung natürlicher Sprache als Lerninhalt in der Schule	18
4 Ableitung und Verortung des Forschungsvorhabens	21
4.1 Ableitung des Forschungsvorhabens	21
4.2 Abgrenzung zu bestehenden Forschungsarbeiten und Lernumgebungen	23
II Lernziele, Design-Prinzipien und Rahmenbedingungen der computergestützten Modellierungsprojekte	27
5 Einleitung	29
6 Übergreifende Lernziele	31
7 Design-Prinzipien zur Erstellung der Lehr- und Lernmaterialien	35
8 Aufbau des Lehr- und Lernmaterials	43
8.1 Aufbau des digitalen Lernmaterials	43
8.1.1 Digitale Arbeitsblätter	44

8.1.2	Tipps	48
8.1.3	Vertiefendes Zusatz- und Informationsmaterial	49
8.1.4	Antwortblätter	50
8.2	Aufbau des begleitenden Lehrmaterials	50
8.2.1	Präsentationsfolien	51
8.2.2	Stundenverlaufsplan und Lösungen	51
9	Formate der Modellierungsprojekte und ihre üblichen Abläufe	53
9.1	Ablauf eines Modellierungstages	53
9.2	Ablauf einer Modellierungs-Doppelstunde	56
9.3	Vergleich der Formate	57
III	Lehr- und Lernmaterial zur Wortvorhersage	59
10	Einleitung	61
11	Theoretischer Hintergrund	63
11.1	<i>N</i> -Gramm-Modelle	63
11.2	Schätzen der Übergangswahrscheinlichkeiten	66
11.2.1	Maximum-Likelihood-Schätzer für das Uni-Gramm-Modell	67
11.2.2	Maximum-Likelihood-Schätzer für das Bi-Gramm-Modell	70
11.3	Kombination der Modelle	70
11.3.1	Rückfallstrategie	71
11.3.2	Interpolation	72
11.4	Bewertung	74
11.5	Schätzen der Interpolationsgewichte	76
11.5.1	Existenz und Eindeutigkeit der Lösung des Optimierungsproblems	76
11.5.2	Lösen des Optimierungsproblems	77
11.6	Stärken und Schwächen des <i>N</i> -Gramm-Modells	78
12	Anwendung der mathematischen Methoden mit realen Textdaten	81
13	Umsetzung im Lernmaterial	85
13.1	Vorwissen und Lernziele	88
13.2	Erläuterung des Lehr- und Lernmaterials	89
13.2.1	Vorüberlegungen zur didaktischen Reduktion des <i>N</i> -Gramm-Modells	89
13.2.2	Arbeitsblatt 1: Wortvorhersage mit dem Bi-Gramm-Modell	91
13.2.3	Arbeitsblatt 2: Uni- und Tri-Gramm-Modell	96
13.2.4	Arbeitsblatt 3: Kombination der <i>N</i> -Gramm-Modelle	98
13.2.5	Arbeitsblatt 4: Bewertung des Sprachmodells	100
13.2.6	Arbeitsblatt 5: Festlegen der Interpolationsgewichte und Diskussion	107
13.2.7	Weiterführendes Material	109
13.2.8	Exkurs: Umsetzung in einer Doppelstunde	114

13.3 Curriculare Anbindung	117
14 Erfahrungswerte und didaktische Reflexion	121
14.1 Bisherige Erfahrungen	121
14.2 Reflexion	129
15 Ausblick	133
 IV Lehr- und Lernmaterial zur Spracherkennung	135
16 Einleitung	137
17 Theoretischer Hintergrund	139
17.1 Vorverarbeitung und Merkmalsextraktion	141
17.1.1 Fouriertransformation	141
17.1.2 Kurzzeitspektralanalyse	142
17.1.3 Filterbank	143
17.2 Klassifikation	145
17.2.1 Abstandsmaß	145
17.2.2 Abstände von Sprachmustern	146
17.2.3 Nächster-Nachbar-Verfahren	148
17.3 Modellverbesserung: zeitliche Anpassung der Sprachmuster	150
17.3.1 Zuordnungsfunktion	150
17.3.2 Lösen des Optimierungsproblems: dynamische Programmierung	153
17.4 Bewertung des Spracherkennungssystems	155
18 Anwendung der mathematischen Methoden mit realen Sprachsignalen	159
19 Umsetzung im Lernmaterial	163
19.1 Vorwissen und Lernziele	165
19.2 Erläuterung des Lehr- und Lernmaterials	166
19.2.1 Arbeitsblatt 1: Einen Ton mathematisch modellieren	167
19.2.2 Arbeitsblatt 2: Die Fouriertransformation am Beispiel von Dreiklängen	168
19.2.3 Arbeitsblatt 3: Merkmale reduzieren	169
19.2.4 Arbeitsblatt 4: Abstände zwischen Merkmalsvektoren	176
19.2.5 Arbeitsblatt 5: Bewertung des Spracherkennungssystems	181
19.2.6 Arbeitsblatt 6: Zeitliche Anpassung der Sprachmuster	186
19.3 Curriculare Anbindung	191
20 Erfahrungswerte und didaktische Reflexion	195
20.1 Bisherige Erfahrungen	195
20.2 Reflexion	201
21 Ausblick	207

V Schlussbetrachtung	211
22 Zusammenfassung und Schlussfolgerungen	213
23 Ausblick	217
Literaturverzeichnis	221
Anhang	
A Lernmaterial zum Thema Wortvorhersage	235
A.1 Arbeitsblatt 1	236
A.2 Arbeitsblatt 2	244
A.3 Arbeitsblatt 3	249
A.4 Arbeitsblatt 4	254
A.5 Arbeitsblatt 5	266
A.6 Arbeitsblatt 5 open	269
A.7 Zusatzblatt 3	272
A.8 Zusatzblatt 4	275
A.9 Zusatzblatt 5	277
A.10 Präsentationsfolien	279
B Lernmaterial zum Thema Spracherkennung	291
B.1 Arbeitsblatt 1	292
B.2 Arbeitsblatt 2	300
B.3 Arbeitsblatt 3	309
B.4 Arbeitsblatt 4	318
B.5 Arbeitsblatt 5	324
B.6 Arbeitsblatt 6	332
B.7 Präsentationsfolien	342
C Evaluationen der Durchführungen	355
C.1 Evaluationen der Durchführungen zum Thema Wortvorhersage	355
C.2 Evaluationen der Durchführungen zum Thema Spracherkennung	373

Abbildungsverzeichnis

2.1	Modellierungskreislauf entnommen aus Blum (1985, S. 200)	5
2.2	Modellierungskreislauf des Projektes CAMMP entnommen aus Frank et al. (2022, S. 2), angelehnt an Blum (1985, S. 200)	6
2.3	Modellierungsspirale entnommen aus Frank et al. (2022, S. 3), angelehnt an die computergestützte Helix, veröffentlicht durch Computer-Based Math (o. D.)	7
2.4	Modellierungskreislauf mit digitalen Medien entnommen aus Greefrath und Siller (2018, S. 12)	10
3.1	Schritte des überwachten Lernens	16
8.1	Bildschirmfoto der Willkommensseite der Lernplattform	44
8.2	Bildschirmfoto eines digitalen Arbeitsblatts aus dem Lernmaterial zu Spracherkennung	45
8.3	Bildschirmfoto einer Ausgabe auf einem digitalen Arbeitsblatt aus dem Lernmaterial zu Spracherkennung	49
8.4	Bildschirmfoto eines Informationsblattes zu den Logarithmusgesetzen aus dem Lernmaterial zur Wortvorhersage	50
10.1	Anwendungsbeispiele unterschiedlicher Assistenzsysteme: Autokorrektur (oben), Wortvervollständigung (mittig), Wortvorhersage (unten)	61
13.1	Übergangsgraph und Übergangstabelle des Bi-Gramm-Modells	93
13.2	Übergangsgraphen zum aus vier Sätzen bestehenden Trainingsdatensatz in verschiedenen Modellen	96
13.3	Bildschirmfoto der Präsentationsfolie zum Vergleich der <i>N</i> -Gramm-Modelle	98
13.4	Rolle der Test- und Trainingsdaten im <i>N</i> -Gramm-Modell	101
13.5	Veranschaulichung der Ermittlung der Modellwahrscheinlichkeit an einem konkreten Testdatensatz	103
13.6	Baumdiagramm zur Ermittlung der Modellwahrscheinlichkeit	105
13.7	Bildschirmfoto der Webanwendung (https://workshops.cammp.online/demo)	115
13.8	Übergangstabelle für die Generierung von Texten	116
14.1	Entwickeltes Optimierungsverfahren einer Lerngruppe	126
17.1	Verarbeitungsschritte bei der Einzelworterkennung, angelehnt an (Fellbaum, 2012, S. 312)	140

17.2 Mel-Filterbank mit 25 Dreiecksfiltern, einer unteren Grenzfrequenz von 100 Hz und einer oberen Grenzfrequenz von 6800 Hz	144
17.3 Euklidischer Abstand zwischen den Punkten A und B im zweidimensionalen Raum	146
17.4 Verletzung der Dreiecksungleichung durch D_{DTW} am Beispiel der Sprachmustern $A = (1, 10, 8, 10)$, $B = (3, 4, 9, 8)$, $C = (9, 4, 9, 2)$	152
17.5 Tabelle für die Zwischenspeicherung der Abstandswerte	154
18.1 Wahrheitstabelle der entwickelten Spracherkennungssysteme	160
19.1 Funktionsweise der Fouriertransformation ausgehend von links Ausgangssignal, mittig Zerlegung und rechts Amplitudenspektrum	169
19.2 Abfolge dreier Dreiklänge in verschiedenen Darstellungen	171
19.3 Bildschirmfoto des Amplitudenspektrums eines Audiosignals in verschiedenen Zeitblöcken, auswählbar durch einen Schieberegler	173
19.4 Bildschirmfoto des Amplitudenspektrums eines spezifischen Zeitblockes mit eingezeichneten Grenzfrequenzen und anhörbaren Audiodateien des originalen und gefilterten Audiosignals	174
19.5 gemitteltes Amplitudenspektrum eines spezifischen Zeitblockes	175
19.6 Amplitudenspektrum eines spezifischen Zeitblockes nach der Mel-Filtrierung	176
19.7 Abstand eines Referenzmusters zu einem Testmuster im Zweidimensionalen im ersten Zeitblock	178
19.8 Gegenüberstellung der Tipps zur Abstandsbestimmung im zwei-, drei- und mehrdimensionalen Raum	180
19.9 Rolle der Test- und Trainingsdaten bei der Klassifizierung von Sprachsignalen	182
19.10 Wahrheitsmatrix für die Klassifikation mit einfachem Mustervergleich	183
19.11 Das erste Merkmal der Muster A und B aufgetragen über den Zeitblöcken	187
19.12 Optimaler Zuordnungspfad der Anpassung der Muster A und B	188
19.13 Beispiele von Zuordnungspfaden, bei denen in (1) die Kontinuitätsbedingung, in (2) die Monotoniebedingung und in (3) die Randwertbedingung verletzt werden	189
20.1 Bildschirmfoto der Aufgabe 4b als Beispiel für den Einsatz eines Überprüfbuttons	197

Tabellenverzeichnis

6.1 Übersicht über die Lernziele in den Modellierungstagen, aufgeteilt auf zentrale Schritte des Maschinellen Lernens, angelehnt an Geisler et al. (under review)	33
8.1 Symbole im Lernmaterial und deren Bedeutung	46
12.1 Kreuzentropie verschiedener Modelle (alle Werte sind auf zwei Stellen gerundet)	82
12.2 Kreuzentropie des geglätteten interpolierten Tri-Gramm-Modells mit Interpolationsgewichten $\tilde{g}_0, \tilde{g}_1, \tilde{g}_2, \tilde{g}_3 = 0, 25$ trainiert mit verschiedenen Datensätzen (alle Werte sind auf zwei Stellen gerundet)	83
13.1 Tabellarische Übersicht über das Lernmaterial zur Wortvorhersage (Dauer in Minuten)	87
14.1 Tabellarische Übersicht über durchgeführte Modellierungsprojekte mit dem Lernmaterial zur Wortvorhersage. In den mit * markierten Veranstaltungen wurde in einem erweiterten Zeitrahmen das Zusatzmaterial zu weiterführenden offenen Problemstellungen bearbeitet.	122
17.1 Wahrheitsmatrix für eine binäre Klassifikation mit n Testdatenpunkten	156
18.1 Qualitätsmaße verschiedener Spracherkennungssysteme	161
19.1 Tabellarische Übersicht über das Lernmaterial zur Spracherkennung (Dauer in Minuten)	164
20.1 Tabellarische Übersicht über durchgeführte Modellierungsprojekte mit dem Lernmaterial zur Spracherkennung	196

Teil I

Theoretischer Rahmen

1 Einleitung

In dieser Arbeit soll gezeigt werden, wie relevante Problemstellungen aus dem Anwendungskontext der Verarbeitung natürlicher Sprache in den Mathematikunterricht integriert werden können. Kapitel 2 stellt daher die mathematische Modellierung als zentrales Werkzeug für das Lösen realer Probleme mithilfe von Mathematik – zunächst losgelöst von dem spezifischen Anwendungskontext NLP – vor. Das Kapitel gibt einen Überblick darüber, was unter mathematischer Modellierung verstanden wird, motiviert die Integration von mathematischen Modellierungsaktivitäten im Mathematikunterricht und zeigt, welche Rolle digitale Medien bei der Vermittlung von mathematischer Modellierung spielen.

Kapitel 3 stellt den Anwendungskontext der entwickelten Modellierungsprojekte, die Verarbeitung natürlicher Sprache, vor. Da dieser Bereich eng mit Künstlicher Intelligenz und Maschinellem Lernen verknüpft ist, werden auch diese Begriffe erläutert und deren Verbindung zur Verarbeitung natürlicher Sprache beleuchtet. Zudem werden Gründe für die Behandlung von Künstlicher Intelligenz und Sprachverarbeitung im schulischen Kontext insbesondere im Mathematikunterricht dargelegt, um das Entwicklungsziel dieser Arbeit zu legitimieren.

Daraus ableitend wird in Kapitel 4 das Forschungsvorhaben dieser Arbeit erläutert, und die Arbeit wird im Bereich der fachdidaktischen Forschung zu Künstlicher Intelligenz, Maschinellem Lernen und Verarbeitung natürlicher Sprache verortet. Bestehende Forschungsarbeiten und Lernumgebungen zu diesen Themen werden vorgestellt, um eine Abgrenzung und Weiterentwicklung im Vergleich zu bisherigen Ansätzen aufzuzeigen. So wird dargelegt, wie diese Arbeit neue Perspektiven und Ansätze für den Unterricht eröffnet und welche spezifischen Beiträge sie zur Integration von Bildung über Künstlicher Intelligenz und Verarbeitung natürlicher Sprache im Mathematikunterricht leistet.

2 Mathematische Modellierung

2.1 Grundlagen der Mathematischen Modellierung

Der Begriff des Modellierens wurde insbesondere von Pollak (1977) geprägt, welcher es als das (wiederholte) Hin- und Herwechseln zwischen Mathematik und dem „Rest der Welt“ bezeichnet (Pollak, 1977, S. 255 - 256). Er legt somit den Fokus auf den *Prozess* des Lösen von Problemen aus der Realität.

Der Modellierungsprozess wird häufig idealisiert als Kreislauf dargestellt, der unterschiedliche Phasen enthält. In der Literatur sind verschiedene Darstellungen des Modellierungskreislaufs zu finden (vgl. Blum, 1985; Blum, 2010; Blum & Leiss, 2005; Ortlieb, 2004). Der vierschrittige Modellierungskreislauf von Blum (1985, S. 200) (vgl. Abbildung 2.1) ist dabei sicherlich einer der bekanntesten. Eine davon leicht abgewandelte Version stellt der Modellierungskreislauf in Abbildung 2.2 dar, welcher später auch in den im Rahmen dieser Arbeit erstellten Lernmaterialien zum Einsatz kommt. Der Prozess des Modellierens soll nun durch die Erläuterung der einzelnen Teilschritte dieses Kreislaufes näher beleuchtet werden.

Ausgangspunkt ist ein Problem aus der Realität (*reales Problem*), welches aufgrund seiner Komplexität auf Basis geeigneter Annahmen zunächst vereinfacht und durch die Auswahl relevanter Informationen strukturiert werden muss, um dieses überhaupt mathematisch beschreiben zu können. Dieser Schritt des *Vereinfachens* liefert das *vereinfachte Problem*. Im Anschluss folgt die Phase des Mathematisierens, in der der Schritt von der realen Welt in die mathematische Welt vollzogen wird. Dabei wird

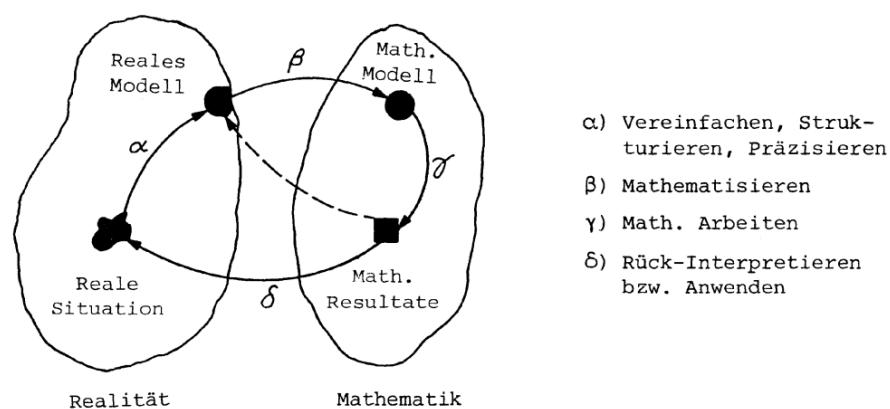


Abbildung 2.1: Modellierungskreislauf entnommen aus Blum (1985, S. 200)

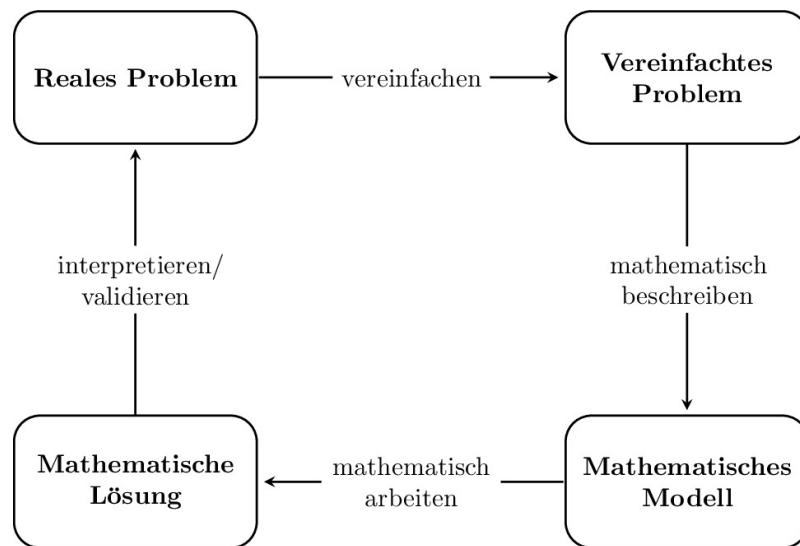


Abbildung 2.2: Modellierungskreislauf des Projektes CAMMP entnommen aus Frank et al. (2022, S. 2), angelehnt an Blum (1985, S. 200)

das vereinfachte Problem *mathematisch beschrieben*, beispielsweise durch das Aufstellen einer Gleichung, womit ein *mathematisches Modell* entwickelt wird. Sobald das mathematische Modell steht, beginnt die Phase des *mathematischen Arbeitens*. Hier wird das Problem mithilfe der symbolischen, formalen und technischen Elemente der Mathematik bearbeitet, um eine Lösung innerhalb der mathematischen Welt zu finden. Dieser Prozess führt im Idealfall zu einer *mathematischen Lösung*. Die gefundene Lösung muss anschließend zurück in die reale Welt übersetzt werden, die mathematische Lösung somit im Bezug auf das reale Problem *interpretiert* werden. Die Ergebnisse werden außerdem *validiert*, was die Überprüfung der Plausibilität der Resultate meint. Ebendiese einzelnen Teilschritte des Modellierungsprozesses spiegeln die von der Kultusministerkonferenz ausgewiesene prozessbezogene Kompetenz des Modellierens wider (Kultusministerkonferenz, 2012, S. 15).

Es ist durchaus häufig der Fall, dass die oben beschriebenen Schritte nicht in der hier angegebenen Reihenfolge durchlaufen werden. Wird beispielsweise bereits beim Mathematisieren klar, dass die notwendigen mathematischen Kompetenzen oder Werkzeuge fehlen, um das vereinfachte Problem mathematisch zu beschreiben, so muss das reale Problem gegebenenfalls stärker vereinfacht werden. Erst dann kann erneut versucht werden, ein mathematisches Modell zu entwickeln (Maaß, 2007, S. 13). Ist die Lösung nach einem ersten Durchlauf noch nicht zufriedenstellend, so wird der Kreislauf erneut durchlaufen und es werden in den einzelnen Teilschritten Veränderungen im Vergleich zum erstmaligen Durchgang vorgenommen. Beispielsweise kann das Problem weniger stark vereinfacht werden oder es können andere mathematische Methoden zur Lösung des mathematischen Problems herangezogen werden. Mit jedem Durchlauf lässt sich eine zunehmend bessere und problemadäquatere Lösung des Problems erreichen. Das Modellieren ist somit als iterativer Prozess zu verstehen, was besonders schön in der Darstellung des Modellierungsprozesses als eine

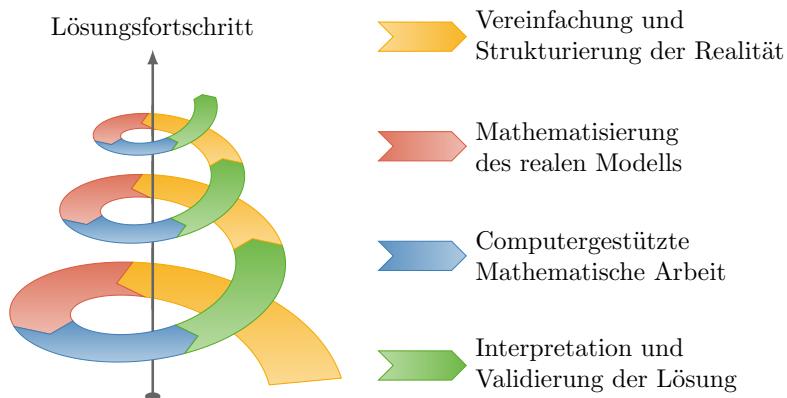


Abbildung 2.3: Modellierungsspirale entnommen aus Frank et al. (2022, S. 3), angelehnt an die computergestützte Helix, veröffentlicht durch Computer-Based Math (o. D.).

sich schließende Spirale ersichtlich wird. Diese Spirale zeigt nämlich zusätzlich zu den Modellierungsschritten auch den Lösungsfortschritt an (vgl. Abbildung 2.3).

Neben den beschriebenen Modellierungskreisläufen, die insbesondere als Orientierungshilfe für Schülerinnen und Schüler in Form eines Metamodells dienen sollen, gibt es auch Darstellungen, die speziell zur kognitiven Analyse und Diagnose von Modellierungsaktivitäten entwickelt wurden. In der Mathematikdidaktik ist der siebenschrittige Modellierungskreislauf nach Blum und Leiß (2005, S. 19) weit verbreitet. Im Vergleich zu den Kreisläufen von Frank et al. (2022, S. 2) und Blum (1985, S. 200) umfasst dieser eine zusätzliche Phase, die sich auf das Verständnis der Realsituation durch die Lernenden konzentriert. Der Kreislauf hat sich zudem als geeignete Grundlage für die Untersuchung des Zusammenspiels von digitalen Medien und mathematischer Modellierung erwiesen und wird daher in Abschnitt 2.3 in einer erweiterten Form erneut aufgegriffen.

2.2 Mathematische Modellierung in der Schule

Modellierungsaktivitäten können im Unterricht in Form von Modellierungsaufgaben umgesetzt werden. Angelehnt an Abschnitt 2.1 werden darunter Aufgaben verstanden, in denen ein reales Problem mit Mathematik gelöst werden soll. Hierbei sind Modellierungsaufgaben von eingekleideten Aufgaben insofern zu unterscheiden, als dass bei eingekleideten Aufgaben die Einkleidung prinzipiell austauschbar ist und der Problemkontext somit nicht vollständig durchdrungen werden muss, um die Aufgabe lösen zu können, wohingegen bei Modellierungsaufgaben eine substantielle Übersetzung zwischen Realität und Mathematik erforderlich ist (Blum, 2007, S. 1).

Das Integrieren von Modellierungsaufgaben im Mathematikunterricht lässt sich insbesondere mit Blick auf die von Heinrich Winter (1995, S. 37) formulierten Grunderfahrungen legitimieren. Zu diesen schreibt Winter Folgendes:

„Der Mathematikunterricht sollte anstreben, die folgenden drei Grunderfahrungen, die vielfältig miteinander verknüpft sind, zu ermöglichen:

- (1) Erscheinungen der Welt um uns, die uns alle angehen oder angehen sollten, aus Natur, Gesellschaft und Kultur, in einer spezifischen Art wahrzunehmen und zu verstehen,
- (2) mathematische Gegenstände und Sachverhalte, repräsentiert in Sprache, Symbolen, Bildern und Formeln, als geistige Schöpfungen, als eine deduktiv geordnete Welt eigener Art kennen zu lernen und zu begreifen,
- (3) in der Auseinandersetzung mit Aufgaben Problemlösefähigkeiten, die über die Mathematik hinaus gehen, (heuristische Fähigkeiten) zu erwerben.

Das Wort Erfahrung soll zum Ausdruck bringen, dass das Lernen von Mathematik weit mehr sein muss als eine Entgegennahme und Abspeicherung von Information, dass Mathematik erlebt (möglicherweise auch erlitten) werden muss.“

(Winter, 1995, S. 37)

Einer der wichtigsten Gründe, weshalb im Mathematikunterricht modelliert werden sollte, ist sicherlich, weil Lernende durch die Beschäftigung mit realen, authentischen Problemstellungen zur Erschließung der Umwelt mit mathematischen Mitteln befähigt werden. Dies entspricht der ersten der drei Winterschen Grunderfahrungen. Um reale Situationen mit mathematischen Mitteln verstehen und bewältigen zu können, müssen geeignete Anwendungen und Modellierungsbeispiele explizit behandelt werden (Blum, 2015, S. 81). Innermathematische Beispiele können diese Funktion aufgrund des fehlenden Transfers nicht leisten. Dafür braucht es Aufgaben, deren Anwendungsbezüge über eingekleidete Aufgaben hinausgehen.

Zudem wird durch Modellierungsaufgaben der Bezug der Mathematik zur Realität gezeigt und damit auch die Relevanz der Mathematik für reale Anwendungen verdeutlicht. Modellierungsaufgaben können daher das Interesse an Mathematik wecken oder steigern (Greefrath et al., 2013, S. 20). Die Behandlung von Modellierungsaufgaben kann außerdem zu einer erhöhten Motivation bei den Lernenden beitragen. Zusätzlich wurde festgestellt, dass mathematische Inhalte, welche beim Modellieren angewandt werden, besser verstanden und behalten werden können (Greefrath et al., 2013, S. 20). Blum (2015, S. 81) bezeichnet dies als die lernpsychologische Rechtfertigung.

Durch die Beschäftigung mit Modellierungsaufgaben können zudem verschiedenste Kompetenzen, wie Problemlösefähigkeiten und Modellierungskompetenzen (Blum, 2015, S. 81) aber auch soziale Kompetenzen (Greefrath et al., 2013, S. 20) gefördert werden. Damit bieten Modellierungsaufgaben das Potential, Lernenden auch die dritte Wintersche Grunderfahrung zu ermöglichen.

Da der Mathematikunterricht ein adäquates Bild der Mathematik als Wissenschaft vermitteln soll, und die Anwendung der Mathematik in der außermathematischen

Welt dazu gehört, ist auch dies ein wichtiger Grund für das Modellieren im Mathe-matikunterricht (Blum, 2015, S. 81).

Die Forderung, die Beziehung zwischen Mathematik und Realität im Unterricht stärker zu betonen, ist ein wiederkehrendes Thema in zahlreichen Publikationen – von den 1960er-Jahren bis hin zu aktuellen Veröffentlichungen, wie die Arbeiten von Pollak (1977), Blum (1985), Blum und Niss (1991) und Vos (2018) zeigen. Auch im aktuellen Bildungsplan Baden-Württembergs sowie in den Bildungsstandards für das Fach Mathematik ist Modellieren als eine der zentralen (prozessbezogenen) Kompetenzen aufgeführt (Kultusministerkonferenz, 2012, S. 11 - 12; Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 6). Dennoch gibt es Hinweise darauf, dass in der Unterrichtspraxis Modellierungsaufgaben meist zu kurz kommen (Jordan et al., 2008, S. 99).

2.3 Mathematische Modellierung und das Zusammenspiel mit digitalen Medien

Digitales Lernmaterial im schulischen Kontext ist aufgrund des fortschreitenden technologischen Wandels der Gesellschaft bereits seit längerem ein viel diskutiertes Thema. Durch die Covid-19-Pandemie und den damit verbundenen Fernunterricht hat es zusätzlich erheblich an Bedeutung für den Schulunterricht gewonnen. Der kompetente und kreative Umgang mit Technologien ist eine Schlüsselqualifikation in der heutigen Zeit, da Schülerinnen und Schüler in ihrem Alltag stetig mit neuen und sich schnell weiterentwickelnden digitalen Medien und Technologien konfrontiert sind. Die Digitalisierungsstrategie des Landes Baden-Württemberg betont daher nicht nur die fächerübergreifende Leitperspektive der Medienbildung, sondern schafft durch eine verbesserte technische Ausstattung die Voraussetzungen dafür, dass der Medieneinsatz in Schulen auch tatsächlich umgesetzt werden kann (Ministerium für Kultus, Jugend und Sport Baden-Württemberg [Ministerium BW], 2023a).

Für die mathematische Modellierung haben digitale Lernmaterialien eine besondere Bedeutung. Sie ermöglichen in vielen Fällen die Lösung realer Probleme im Schulkontext überhaupt erst und können gerade bei komplexen realen Problemen eine sinnvolle Unterstützung für die Lernenden sein (Greefrath & Siller, 2018, S. 9 - 10).

Prinzipiell gibt es zwei grundlegende theoretische Modelle, die die Nutzung digitaler Medien in die Theorie des mathematischen Modellierens einbetten. Das erste eignet sich insbesondere dann, wenn das digitale Medium als digitales Werkzeug in Schülerhand genutzt wird. In diesem Modell wird in einem erweiterten Modellierungskreislauf die Ebene der digitalen Werkzeuge neben der Ebene der Mathematik und der Ebene der Realität berücksichtigt. Hierbei wird der Fokus auf die notwendige Übersetzung zwischen dem mathematischen Modell und einem Modell für das digitale Werkzeug gelegt (Greefrath & Siller, 2018, S. 11). Das kann beispielsweise die Darstellung mathematischer Methoden im Programmcode beinhalten.

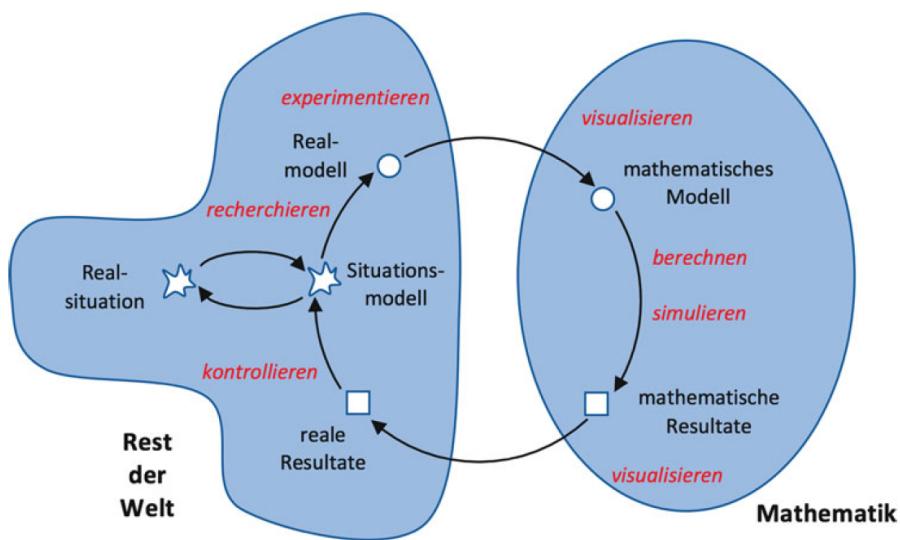


Abbildung 2.4: Modellierungskreislauf mit digitalen Medien entnommen aus Greefrath und Siller (2018, S. 12)

Eine zweite Betrachtungsweise berücksichtigt den Einsatz digitaler Medien in allen Teilprozessen des Modellierungskreislaufs. Hierbei sind sowohl digitale Werkzeuge als auch digitale Lernumgebungen inbegriffen. Mit digitalen Lernumgebungen werden dabei nach Roth (2022, S. 116) Lernumgebungen bezeichnet, in welche mit einem digitalen Werkzeug erstellte Applets eingebunden sind. Abbildung 2.4 visualisiert, an welchen Stellen des Modellierungskreislaufs digitale Medien für welche Tätigkeiten eingesetzt werden können. Inwiefern digitale Medien in den einzelnen Teilprozessen unterstützen können, wurde bereits von Greefrath und Siller (2010, 2018) dargelegt und soll hier noch einmal kurz zusammengetragen werden:

- Recherchieren und Experimentieren: Digitale Medien können besonders beim Recherchieren und Experimentieren für das Entdecken mathematischer Zusammenhänge förderlich sein (Hischer & Lambert, 2002, S. 116 - 117). So kann beispielsweise eine reale Situation in ein geometrisches Modell mit dynamischer Geometriesoftware übertragen werden und darin experimentiert werden. Zudem kann eine Recherche im Internet nach Informationen zum Anwendungskontext hilfreich sein, um das Problem zu verstehen sowie Annahmen und Vereinfachungen treffen zu können.
- Visualisieren: Der Einsatz von Technologien erleichtert Visualisierungen bei der Datenexploration, aber auch das Visualisieren der mathematischen Resultate (Siller & Greefrath, 2010, S. 2139). Bei realen Problemen ist aufgrund der großen Datenmengen eine Visualisierung mit digitalen Werkzeugen häufig unabdingbar.
- Berechnen und Simulieren: Durch den Einsatz digitaler Werkzeuge und digitaler Lernumgebungen können repetitive oder besonders zeitintensive und aufwändige Berechnungen an den Computer ausgelagert werden (Greefrath & Siller, 2018, S. 5). Dadurch kann der Fokus auf den Modellierungsprozess ge-

legt werden. Auch komplexere mathematische Verfahren, die in der jeweiligen Klassenstufe noch nicht bekannt sind, lassen sich so als „Black Box“ verwenden. Dies ermöglicht es, anspruchsvolle Fragestellungen zu bearbeiten und gibt den Lernenden einen realistischen Einblick in den Anwendungsprozess, ohne dass sie durch die technischen Details überfordert werden. Der Einsatz digitaler Medien macht so viele interessante Problemstellungen aus der realen Welt aufgrund deren Komplexität und der Arbeit mit großen Datenmengen überhaupt erst für den Schulunterricht zugänglich (Siller & Greefrath, 2010, S. 2138; Greefrath & Siller, 2018, S. 5). Zudem wird exploratives Arbeiten zum Beispiel in Form von Parameterstudien und systematischen Variationen durch den Einsatz digitaler Medien in einem vertretbaren Zeitrahmen ermöglicht.

- **Kontrollieren:** Digitale Medien können das Kontrollieren und Überprüfen mathematischer Resultate insbesondere durch grafische Visualisierungen und das einfache Berechnen von Qualitätsmaßen unterstützen (Greefrath & Siller, 2018, S. 10).

Auch in Wissenschaft, Industrie und Forschung werden reale Probleme mit technischen Hilfsmitteln angegangen und gelöst. Die Nutzung digitaler Werkzeuge ist somit nicht nur hinsichtlich der oben genannten Unterstützungen, die das digitale Werkzeug beim Modellierungsprozess bietet, sinnvoll, sondern gibt auch einen authentischen praxisorientierten Einblick in den Ablauf von Problemlösungsprozessen in der Arbeitswelt.

3 Verarbeitung natürlicher Sprache und Künstliche Intelligenz

3.1 Verarbeitung natürlicher Sprache

Neben der mathematischen Modellierung spielt der Anwendungskontext der Verarbeitung natürlicher Sprache für diese Arbeit eine zentrale Rolle. Dieser Abschnitt gibt deshalb einen Überblick darüber, was unter dem Begriff *Verarbeitung natürlicher Sprache* verstanden wird. Eine präzise und abschließende Definition wird jedoch nicht angestrebt, da selbst in der Fachliteratur keine einheitliche Begriffsdefinition existiert. Stattdessen dient dieses Kapitel dazu, die grundlegenden Ideen und Anwendungen der Sprachverarbeitung zu skizzieren. In den Teilen **III** und **IV** dieser Arbeit werden spezifische Unterthemen der Verarbeitung natürlicher Sprache vertieft, wobei jedes dieser Themen erneut mit einer kurzen Einführung einhergeht. Ziel dieses Kapitels ist es, eine erste Orientierung zu bieten und eine Grundlage für die weiterführenden Inhalte der Arbeit zu schaffen.

Das Forschungsfeld der Verarbeitung natürlicher Sprache (engl. Natural Language Processing (NLP)) befasst sich mit der Erkennung und Verarbeitung von menschlicher Sprache durch den Computer (Grigoleit, 2019, S. 1). Unter natürlicher Sprache wird dabei Sprache verstanden, die von Menschen zur Kommunikation untereinander verwendet wird. Die zu analysierenden Äußerungen können sowohl schriftlicher als auch mündlicher Natur sein und aus jeder Sprache stammen. Die Verarbeitung umfasst verschiedene Aufgaben wie die Analyse von Texten, die Erfassung von Äußerungen schriftlicher oder mündlicher Form, sowie die schriftliche und auditive Kommunikation (Nugues, 2006, S. 2). Typische Anwendungen sind beispielsweise Spracherkennungssysteme, maschinelle Übersetzer, Dialogsysteme sowie Rechtschreib- und Grammatikprüfprogramme.

Das Forschungsfeld NLP hat sich als Überschneidung der Bereiche Künstliche Intelligenz (vgl. Abschnitt 3.2) und Linguistik in den 1950er Jahren entwickelt (Nadkarni et al., 2011, S. 1). Ein Großteil der Entwicklungen im Bereich der natürlichen Sprachverarbeitung hat ihren Ursprung in den zu dieser Zeit entstandenen Arbeiten zur maschinellen Übersetzung (Dale et al., 1998, S. 1). Obwohl der gesamte Bereich als Sprachverarbeitung bezeichnet wird, lassen sich zwei Schwerpunkte unterscheiden: das Verstehen von Sprache und das Generieren von Sprache (Dale et al., 1998, S. 1). Sprachverständnis zielt darauf ab, Sprache zu analysieren und eine sinnvolle Repräsentation von Sprache zu erstellen, während Spracherzeugung darauf abzielt, aus einer solchen Repräsentation Sprache zu generieren. Während die Sprachverarbeitung der

Rolle eines Lesers oder Zuhörers entspricht, übernimmt die Spracherzeugung die Funktion eines Schreibers oder Sprechers. Trotz dieser unterschiedlichen Zielsetzungen basieren Methoden beider Teilbereiche häufig auf ähnlichen Ansätzen. Zudem erfordern zahlreiche NLP-Systeme die Kombination von Aufgaben aus beiden Bereichen. Um Texte zu übersetzen muss beispielsweise zunächst die Eingabe des Nutzers analysiert und daraufhin eine passende Übersetzung generiert werden.

3.2 Künstliche Intelligenz und Maschinelles Lernen

Methoden aus dem Bereich der Künstlichen Intelligenz spielen bei der Verarbeitung natürlicher Sprache eine wichtige Rolle. Dieser Abschnitt gibt daher einen Überblick über die Begriffe *Künstliche Intelligenz* und *Maschinelles Lernen* und zeigt die Verzahnung von Künstlicher Intelligenz und Verarbeitung natürlicher Sprache auf. Eine präzise Definition der Begriffe ist nicht das Ziel dieser Arbeit, zumal selbst in der Fachliteratur Uneinigkeit über deren genaue Abgrenzung besteht. In diesem Abschnitt sollen jedoch die beiden Begriffe und ihre Verwendung in der Literatur kurz erläutert werden und ihre Beziehung zur Verarbeitung natürlicher Sprache dargelegt werden. Zudem werden zentrale Aspekte hervorgehoben, die im Zusammenhang mit diesen Konzepten immer wieder eine Rolle spielen.

Künstliche Intelligenz

Der Begriff „Künstliche Intelligenz“ (KI, engl. Artificial Intelligence) wurde erstmals im Jahr 1955 in einem Vorschlag für die Inhalte eines Sommerprojekts am Dartmouth College erwähnt (McCarthy et al., 1955). Bereits dort wurden Themen studiert, die auch heute noch zentrale Bestandteile der KI-Forschung darstellen. Dazu zählen beispielsweise die Automatisierung, das Auslesen, Verstehen und Erzeugen von Texten durch Computer und neuronale Netzwerke. McCarthy et al. (1955) wollten mit diesem Sommerprojekt an der Vermutung forschen, dass „jeder Aspekt des Lernens oder jedes anderen Merkmal der Intelligenz prinzipiell so genau beschrieben werden kann, dass man eine Maschine bauen kann, die es simuliert“¹ [Übers. d. Verf.] (McCarthy et al., 1955, S. 2).

Trotz der langen Forschungsgeschichte existiert bis heute keine allgemein anerkannte Definition des Begriffs „Künstliche Intelligenz“. Dies liegt unter anderem daran, dass der zugrundeliegende Begriff der „Intelligenz“ selbst nicht eindeutig gefasst ist und je nach disziplinärem Kontext unterschiedlich interpretiert wird (D. G. Myers, 2014, S. 400). In der Praxis bearbeiten KI-Systeme heute ein breites Spektrum an Aufgaben, die sich in die fünf Aufgabenbereiche

- Deduktionssysteme und maschinelles Beweisen;
- Wissensbasierte Systeme;

¹ „The study is to proceed on the basis on conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described, that a machine can be made to simulate it.“

- Robotik;
- Musteranalyse und Mustererkennung;
- Intelligente multimodale Mensch-Maschine-Interaktion wie Analyse und Verstehen von Sprache, Bildern und anderen Formen menschlicher Interaktion

einteilen lassen (Bundesregierung Deutschland, 2018, S. 5).

Bereits hier wird die enge Verknüpfung von KI und NLP ersichtlich, da die Verarbeitung natürlicher Sprache als ein Aufgabenbereich des Begriffs künstlicher Intelligenz ausgewiesen wird. Auch ein Blick auf die Inhalte des Sommerprojekts zeigt, dass NLP bereits seit der Begriffsneuschöpfung von KI ein wichtiger Aufgabenbereich zu sein scheint. So heißt das zweite der vorgestellten KI-Probleme „How Can a Computer be Programmed to Use a Language“ (McCarthy et al., 1955, S. 2).

Sehr bekannt im Zusammenhang mit den Begriffen KI und NLP sind auch die Vorarbeiten von Alan Turing. Turing stellte in seiner Veröffentlichung „Computing Machinery and Intelligence“ (Turing, 1950) die Frage, ob Maschinen denken können. Um die Schwierigkeit zu umgehen, Begriffe wie „Denken“ und „Intelligenz“ definieren zu müssen, schlägt er vor, die Frage allein durch Beobachtung eines Experiments zu beantworten. Dieses Experiment ist auch als *Turing-Test* bekannt (Nebel, 2019, S. 304). Der Test beschreibt ein Szenario, in dem ein menschlicher Beobachter schriftlich mit zwei Gesprächspartnern kommuniziert: einem Menschen und einer Maschine. Gelingt es dem Beobachter nicht, zu identifizieren, welcher Gesprächspartner Mensch, welcher Maschine ist, so gilt die Maschine nach Turings Ansatz als „intelligent“.

Zu Turings Zeit war der Test lediglich ein Gedankenexperiment, da die technische Hardware noch zu eingeschränkt war, um solche Systeme zu realisieren. Dennoch hat der Turing-Test die Entwicklung der KI geprägt. Hayes und Ford (1995, S. 972) warnen allerdings davor, die Definition des Begriffes KI am erfolgreichen Bestehen des Turing-Tests fest zu machen. Als Grund nennen sie unter anderem, dass sich der Turing-Test nur auf die maschinelle Sprachverarbeitung fokussiert und andere Felder der KI, wie etwa Computervision oder Robotik, unberücksichtigt lässt (Hayes & Ford, 1995, S. 972 - 973). Letztlich überprüft der Test ausschließlich, ob ein KI-System eine ganz bestimmte Aufgabe so ausführen kann, dass die Ergebnisse nicht von denen eines Menschen zu unterscheiden sind. Dabei wird jedoch nur diese eine Aufgabe aus der Vielzahl möglicher Tätigkeiten betrachtet, die ein KI-System potenziell übernehmen könnte. Dass diese Aufgabe aus der Sprachverarbeitung stammt, zeigt allerdings auch, dass Turing die natürlichsprachliche Kommunikation als zentralen Aufgabenbereich der KI betrachtet.

Um KI-Systeme zu entwickeln werden unterschiedliche Methoden eingesetzt. Diese Methoden lassen sich in regelbasierte Methoden und maschinelle Lernmethoden unterscheiden. Ein grundlegender Unterschied besteht in der Rolle, die die Daten bei den verschiedenen Methoden einnehmen. Bei regelbasierten Methoden spielen Daten eine untergeordnete Rolle. Es werden Entscheidungsregeln festgelegt, welche auf Basis von Wissen konstruiert werden. Maschinelle Lernmethoden sind hingegen datengetrieben (Döbel et al., 2018, S. 11). Dieser Aspekt wird bei der Vorstellung der unterschiedlichen Klassen von maschinellen Lernmethoden deutlich.

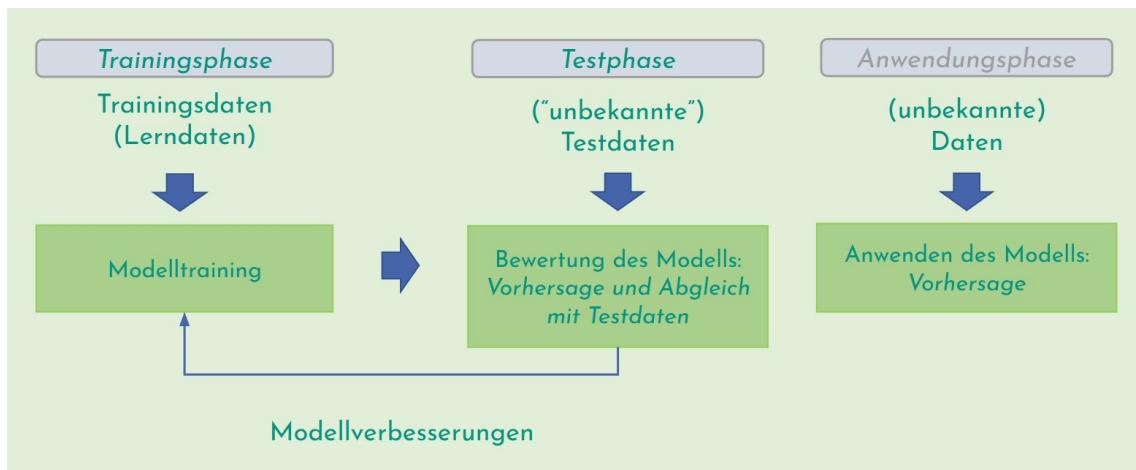


Abbildung 3.1: Schritte des überwachten Lernens

Maschinelles Lernen

Das Maschinelle Lernen (ML, engl. Machine Learning) lässt sich in die drei Klassen von Methoden

- *bestärkendes Lernen* (engl. reinforcement learning)
- *unüberwachtes Lernen* (engl. unsupervised learning)
- *überwachtes Lernen* (engl. supervised learning)

unterteilen (Biehler et al., 2024, S. 3; Hazzan & Mike, 2023, S. 203; Heesen, 2023, S. 19 - 24).

Das Vorgehen beim bestärkenden, unüberwachten und überwachten Lernen wird anhand eines *Klassifizierungsproblems* vorgestellt. Bei einem Klassifizierungsproblem geht es darum eine Zuordnungsfunktion

$$f : \mathbb{R}^n \rightarrow \{K_1, \dots, K_m\}$$

zu finden, die Datenpunkte $x_i \in \mathbb{R}^n$ mit $i = 1, \dots, N$ in eine endliche Menge von m verschiedenen Klassen $\{K_1, \dots, K_m\}$ einteilt. Das Ziel besteht darin möglichst viele Datenpunkte korrekt zuzuordnen. Welche ML-Klasse zur Bestimmung der Zuordnungsfunktion geeignet ist, hängt von der verfügbaren Information ab.

Beim bestärkenden Lernen wird die Rückmeldung zu einer Aktion genutzt, um zukünftige Entscheidungen zu verbessern. Im Kontext des Klassifizierungsproblems bedeutet dies, dass die über die Zuordnungsfunktion f ermittelten Klassenzugehörigkeiten $f(x_i)$ mehrerer Datenpunkte extern bewertet werden. In einem nächsten Schritt werden die Bewertungsdaten dafür genutzt, die Zuordnungsfunktion und damit die Entscheidungsregel anzupassen.

Das unüberwachte Lernen hat das Ziel, Muster und Zusammenhänge in Daten zu finden. Zur Verfügung stehen allerdings nur die Eingangsdaten und kein zugehöriger Output. Wird erneut das Beispiel eines Klassifizierungsproblems betrachtet, so be-

deutet dies, dass zwar Datenpunkte x_i , aber nicht deren Klassenzugehörigkeit K_{x_i} vorliegen. Das Ziel ist es, die Eingangsdaten in geeignete Klassen zu unterteilen.

Im Gegensatz dazu liegen beim überwachten Lernen Eingangsdaten vor, deren zugehöriger Output gegeben ist. Wird wieder das Beispiel eines Klassifizierungsproblems betrachtet, so werden beim überwachten Lernen Eingangsdaten x_i für die Entwicklung eines Klassifizierungsalgorithmus genutzt, deren Klassenzugehörigkeit K_{x_i} (Output) bereits bekannt ist. Diese Daten werden auch als *Trainingsdaten* und das Auswerten der Trainingsdaten als *Lernen* oder *Training* bezeichnet (vgl. Abbildung 3.1, Trainingsphase). Aus mathematischer Sicht handelt es sich beim Training um die Bestimmung von Parametern eines gewählten Modellansatzes (Biehler et al., 2024, S. 3). Das Modell, welches bei Klassifizierungsproblemen durch die Zuordnungsfunktion f repräsentiert wird, soll den Zusammenhang zwischen bekannten Eingangsdaten und der Klassenzugehörigkeit bestmöglich beschreiben. Anschließend werden sogenannte *Testdaten* (y, K_y) mit $y \in \mathbb{R}^n$ und $K \in \{K_1, \dots, K_m\}$ genutzt, um zu validieren, wie gut das Modell für die Anwendung auf Daten, die nicht beim Training verwendet wurden, geeignet ist. Bei der Bewertung des Modells, auch *Testphase* genannt wird das Modell auf die Testdaten angewandt und die über die Zuordnungsfunktion ermittelte Klassenzugehörigkeit $f(y)$ mit der tatsächlichen Klassenzugehörigkeit K_y verglichen (vgl. Abbildung 3.1, Testphase). Ziel ist es, das gelernte Modell schließlich zur Klassifizierung von Daten, die sich noch nicht im Trainings- oder Testdatensatz befinden und deren Klassenzugehörigkeit tatsächlich unbekannt ist, anwenden zu können (vgl. Abbildung 3.1, Anwendungsphase). Insgesamt wird somit eine datengetriebene mathematische Modellierung vorgenommen.

Beispiele für Methoden des überwachten Lernens sind unter anderen *Support Vector Machines* und *Nächster-Nachbar-Verfahren* (Döbel et al., 2018, S. 10). Bei Support Vector Machines wird die Zuordnungsfunktion durch Festlegen einer Hyperebene formuliert. Diese Hyperebene, trennt im Falle eines Zwei-Klassen-Klassifizierungsproblems die Klassen so, dass der Abstand zwischen der separierenden Hyperebene und den nächstgelegenen Datenpunkten beider Klassen maximiert wird. Beim Nächster-Nachbar-Verfahren wird ein zu klassifizierender Datenpunkt der Klasse des nächstliegenden Datenpunkts zugeordnet. Das Verfahren ist allerdings insofern ein Spezialfall unter den hier genannten Methoden, als dass das Training nicht vorgelagert, sondern zeitgleich mit der Anwendung des Modells auf Testdaten oder gänzlich neuen Daten stattfindet. Die Zuordnungsfunktion wird somit bei jeder Anwendung erneut ermittelt. Eine ausführlichere Behandlung des Nächster-Nachbar-Verfahren erfolgt in Teil IV dieser Arbeit.

Auch sogenannte *N-Gramm-Modelle*, die zur Vorhersage eines Wortes basierend auf den bereits geschriebenen Wörtern (der sogenannten Wortvorgeschichte) verwendet werden, lassen sich unter einem gewissen Blickwinkel dem überwachten Lernen zuordnen. In diesem Kontext dienen die Wortvorgeschichten als Eingabedaten und das jeweils folgende Wort als Output. Die Vorhersage des nächsten Wortes erfolgt anhand einer Wahrscheinlichkeitsverteilung deren Parameter auf Basis von Trainingsdaten geschätzt werden. Eine tiefergehende Behandlung von N-Gramm-Modellen erfolgt in Teil III.

Diese Arbeit beschränkt sich auf Methoden des überwachten Lernens zur Lösung von Problemstellungen aus dem NLP. Bei Problemen aus dem NLP sind die dabei verwendeten Trainings- und Testdaten zumeist geschriebene oder gesprochene Sprachelemente.

3.3 Künstliche Intelligenz und Verarbeitung natürlicher Sprache als Lerninhalt in der Schule

Künstliche Intelligenz als Lerninhalt in der Schule

In der heutigen digitalen Welt gewinnt KI zunehmend an Bedeutung in allen Bereichen des Lebens (zum Beispiel in der Wissenschaft, der Wirtschaft, der Kunst und der Bildung). Dies liegt vor allem an der steigenden Menge an verfügbaren Daten, aber auch der gestiegenen Rechenleistung, sowie der großen Nachfrage der Gesellschaft nach automatisierten Lösungen für verschiedenste Anwendungen (Bundesregierung Deutschland, 2018, S. 10). Schülerinnen und Schüler interagieren täglich mit KI-Systemen wie intelligenten Sprachassistenten (z. B. Siri und Alexa), Chatbots (z.B. ChatGPT) oder personalisierten Empfehlungssystemen (z.B. beim Online-Shopping oder der Auswahl von Streaming-Inhalten auf Plattformen wie Netflix oder Spotify). Gleichzeitig hat KI bereits tiefgreifenden Einfluss auf die Berufswelt genommen und wird, da Unternehmen immer mehr auf KI-Technologien zurückgreifen, in Zukunft eine noch bedeutendere Rolle spielen.

Angesichts dieser Entwicklung ist es essenziell, Heranwachsende darauf vorzubereiten, nicht nur als verantwortungsvolle Nutzende, sondern auch als aktive Mitgestaltende von KI-Systemen zu agieren. Um eine breite Bildung in diesem Bereich zu ermöglichen, sollte KI als Lerninhalt im schulischen Kontext etabliert werden. Diese Forderung wurde bereits oft formuliert (Evangelista et al., 2018, S. 5; Long & Magerko, 2020, S. 1; Mertala & Fagerlund, 2023, S. 12 - 13; Miao & Shiohira, 2024, S. 3; Opel et al., 2019, S. 285; Tedre et al., 2021, S. 110569; Touretzky et al., 2019, S. 9798; Vartiainen et al., 2021, S. 1). Auch die intensive Forschung zum Lehren und Lernen von ML im Schulunterricht (Sanusi et al., 2023, S. 5986) zeigt, dass sich das Forschungsfeld einig ist, dass KI und insbesondere deren Funktionsweise als Lerninhalt im schulischen Unterricht adressiert werden sollte. Der Aufbau eines grundlegenden Verständnisses der Funktionsweise von KI-Systemen ist unerlässlich, um Chancen, Risiken und Gestaltungsmöglichkeiten von KI einschätzen zu können und damit die Lernenden nicht nur für den Umgang mit KI-Systemen im Alltag sondern auch für die zukünftige Berufswelt vorzubereiten.

Eine Begründung, KI als Bildungsinhalt zu identifizieren, lässt sich auch aus dem Bildungskonzept von Wolfgang Klafki (1995) ableiten. KI kann aufgrund seiner aktuellen und zukünftigen Bedeutsamkeit für die Gesellschaft in kultureller, sozialer, wirtschaftlicher und wissenschaftlicher Hinsicht als wesentlicher Inhalt der Allgemeinbildung des gegenwärtigen Zeitalters, von Klafki (1995, S. 99) auch als „epochaltypisches Schlüsselproblem“ bezeichnet, gezählt werden (Tulodziecki, 2020, S. 374). Laut Klafki (1995, S. 99) sollen Bildungsinhalte angepasst an diese Schlüsselprobleme vermittelt werden.

Um KI-Systeme nicht nur anwenden zu können, sondern auch deren Funktionsweise zu verstehen, ist die Auseinandersetzung mit konkreten Methoden des Maschinellen Lernens notwendig. Sie ermöglicht es zu erkennen, dass KI-Systeme auf bereits erfassten Daten basieren und allein aus diesen Schlüsse ziehen und Verallgemeinerungen auf unbekannte Daten durchführen. Ein solcher Einblick hinter die Fassade von KI-Anwendungen, wie etwa den scheinbar empfindungsfähigen und emotional komplexen Persönlichkeiten von Sprachassistenten wie Siri oder Alexa, soll ein fundiertes und kritisches Verständnis dieser Technologien fördern.

KI als Lerninhalt im Mathematikunterricht

KI-Bildung kann und sollte dabei zu einem großen Teil im Mathematikunterricht stattfinden (Bauer et al., 2023, S. 52; Biehler et al., 2024, S. 3; Opel et al., 2019, S. 285; Schönbrodt et al., 2022, S. 75). Ein Hauptgrund ist die inhaltliche Nähe zur Mathematik. Die Entwicklung von KI-Systemen erfordert ein umfangreiches Wissen und Kompetenzen im Fach Mathematik (Hazzan & Mike, 2023, S. 203; Miao & Shiohira, 2024, S. 34). Dies ist darauf zurückzuführen, dass es sich beim Maschinellen Lernen letztlich um datengetriebene mathematische Modellierung handelt. Dabei werden die Modellaussagen im Vergleich zu einer rein deskriptiven Modellierung auf ungewohnte Daten verallgemeinert, um Vorhersagen treffen zu können. KI-Systeme basieren somit im Kern auf mathematischen Methoden. Aus dieser Sichtweise ergeben sich zwei naheliegende Anknüpfungspunkte an den Mathematikunterricht: der Umgang mit Daten und die mathematische Modellierung.

Für die Entwicklung eines KI-Systems bedarf es (gegebenenfalls) der Sammlung, der Exploration, der Aufbereitung, der Analyse von Daten sowie der Interpretation der Datenanalyse und der Prognose aufgrund von Daten. Werden Lernende im Zusammenhang mit der Thematisierung von KI mit dem Umgang mit Daten geschult, so kann deshalb der Aufbau einer umfassenden Datenkompetenz im Sinne einer Data Literacy (Schüller et al., 2019, S. 16) unterstützt werden.

Die Thematisierung von KI als Lerninhalt ermöglicht es außerdem, mathematische Konzepte in einem relevanten und hochaktuellen Kontext anzuwenden und bietet so die Gelegenheit zur Modellierung realer Probleme im Mathematikunterricht (Schönbrodt et al., 2022, S. 74). Dies kann helfen, die Motivation der Lernenden für Mathematik zu steigern, indem die Relevanz der Mathematik für KI-Systeme dargelegt wird und ein Kontext behandelt wird, welcher derzeit Gegenstand des gesellschaftlichen Interesses ist.

Ein Einblick in die Funktionsweise von KI-Systemen zeigt Lernenden, dass solche Systeme auf mathematischen Methoden basieren. Die verwendeten Methoden sind oftmals, sofern sie geeignet didaktisch reduziert werden, mit mathematischem Schulwissen zugänglich und können von Schülerinnen und Schülern nachvollzogen werden. Die Aufklärung über die entscheidende Rolle der Mathematik bei der Entwicklung von KI-Systemen kann dazu beitragen, Missverständnisse über die Natur und Fähigkeiten von KI-Systemen zu vermeiden. Auch Mertala und Fagerlund (2023, S. 12 - 13) sprechen sich für einen Aufbau eines tieferen Verständnisses über die Funktionsweise von KI-Systemen aus um typischen Fehlvorstellungen wie der Vorstellung einer Anthropomorphen KI, einem KI-System mit vorinstalliertem Wissen sowie einem nicht-technologischen KI-System entgegenzuwirken.

Zudem kann die Thematisierung von KI als Lerninhalt im Mathematikunterricht helfen, innermathematisch vernetzend zu denken. KI-Anwendungen basieren auf verschiedenen mathematischen Disziplinen wie Statistik, Wahrscheinlichkeitsrechnung, linearer Algebra und Optimierung. Die Behandlung dieser Themen im Zusammenhang mit KI kann helfen, die Verbindungen zwischen diesen mathematischen Bereichen zu erkennen.

Insgesamt ist die Thematisierung von KI im Mathematikunterricht somit aufgrund der inhaltlichen Nähe nicht nur naheliegend, sondern bietet auch vielfältige Chancen für den Mathematikunterricht.

NLP als Lerninhalt im Mathematikunterricht

Insbesondere NLP als Anwendungsbereich der KI-Forschung eignet sich für eine Thematisierung im Mathematikunterricht. Ein wesentlicher Grund dafür ist die enge Verbindung zwischen NLP und mathematischen Konzepten. Obwohl NLP oft mit sprachwissenschaftlichen Forschungsgebieten und der Informatik in Verbindung gebracht wird, so spielt dennoch die Mathematik eine wichtige Rolle. Um die Verarbeitung von Sprache durch den Computer zu ermöglichen, müssen mathematische Modelle von sprachlichen Strukturen entwickelt werden (Nugues, 2006, S. 1).

NLP wird zudem als äußerst wichtiger Teilbereich der KI angesehen. Viele Fachpublikationen zu KI können dem Teilgebiet NLP zugeschrieben werden (Buchkremer, 2020, S. 38 - 39), und auch die derzeit viel diskutierten Chatbots (u.a. ChatGPT, Perplexity) sind eine Anwendung des NLP. NLP zählt außerdem zu einem der Teilbereiche, der bereits seit der Begriffsneuschöpfung von KI unter diesem Begriff gefasst wurde. Der darauffolgende Aufschwung von KI-Technologien ist auf die Notwendigkeit der Analyse unstrukturierter Informationen insbesondere der Analyse unzähliger Textdaten zurückzuführen und somit ebenfalls zu einem großen Teil dem Anwendungsgebiet des NLP zuzurechnen (Buchkremer, 2020, S. 35).

Die Thematisierung dieses Teilbereichs im Unterricht ist besonders deshalb relevant, weil sich insbesondere die Fehlvorstellung einer anthropomorphen KI bei Lernenden durch die natürlichsprachlichen Interaktion mit Sprachassistenten wie Alexa und Siri ergeben (Kim et al., 2023, S. 9828; Mertala & Fagerlund, 2023, S. 9). Über die Funktionsweise dieser NLP-Technologien aufzuklären erscheint deshalb notwendig, um ein korrektes Verständnis von KI im Allgemeinen zu vermitteln. Dies wird auch durch die Formulierung der von Touretzky et al. (2019) aufgestellten großen Ideen, die alle von Klassenstufe 1 bis 12 kennen sollten, bekräftigt. NLP taucht an vielen Stellen dieser Arbeit aber insbesondere im Zusammenhang mit einer ihrer „Five Big Ideas“, die alle Lernende kennen sollen, auf. Diese Idee besagt, dass die Gestaltung von KI-Systemen, die komfortabel mit Menschen interagieren, eine große Herausforderung für KI-Entwicklung darstellt (Touretzky et al., 2019, S. 9797). Lernende sollen einordnen können, welche Art von Anfragen in natürlicher Sprache von einem KI-System verstanden werden können und verstehen, wie das Analysieren von Sätzen sowie das Generieren von Text funktioniert (Touretzky et al., 2019, S. 9798). In einer ihrer weiteren großen Ideen stellen sie zudem die Tatsache, dass KI-Systeme Sensoren für die Wahrnehmung der Außenwelt nutzen, als besonders wichtig heraus (Touretzky et al., 2019, S. 9797). Sie nennen in diesem Zusammenhang die Spracherkennung als ein Beispiel aus dem NLP Kontext (Touretzky et al., 2019, S. 9798).

4 Ableitung und Verortung des Forschungsvorhabens

4.1 Ableitung des Forschungsvorhabens

In Abschnitt 2.2 wird dargelegt, weshalb Modellierungsaufgaben in den Mathematikunterricht integriert werden sollen. Zudem wird ein Hinweis benannt, der nahelegt, dass diese Aufgaben im Regelunterricht bislang nur einen begrenzten Stellenwert einnehmen. Abschnitt 3.3 beleuchtet die Bedeutung der Thematisierung von KI und der Funktionsweise von KI-Systemen im schulischen Kontext, insbesondere im Mathematikunterricht. Darüber hinaus wird im nachfolgenden Abschnitt (vgl. Abschnitt 4.2) verdeutlicht, dass die Entwicklung von Lernmaterialien für den Mathematikunterricht, welche Maschinelles Lernen als Lerninhalt aufgreifen, bislang kaum vorangetrieben wurde.

Aus der Notwendigkeit eines stärker realitätsbezogenen Unterrichts und des Bedarfs an Entwicklungsforschung im Bereich der KI-Bildung ergibt sich das Ziel dieser Arbeit: die Entwicklung von Lernmaterialien für Modellierungsprojekte, die die Funktionsweise von KI und Methoden des ML anhand realer Problemstellungen einführt. Dabei soll insbesondere NLP als wichtiger Anwendungsbereich von KI thematisiert werden (vgl. Abschnitt 3.3). Zielgruppe der Modellierungsprojekte sind Kurse und Klassen der gymnasialen Mittel- und Oberstufe.

Das Lernmaterial soll somit zwei zentrale Ziele verfolgen. Auf der einen Seite soll den Lernenden durch die Bearbeitung des Lernmaterials ermöglicht werden, die Mathematische Modellierung als Metakonzept für die mathematische Herangehensweise an reale Probleme kennenzulernen. Dabei soll bewusst gemacht werden, welche Relevanz die Mathematik für Probleme aus der Realität hat und wie sie systematisch eingesetzt werden kann, um reale Probleme zu lösen. Exemplarisch soll die Lösung eines realen Problems aus dem NLP-Kontext betrachtet werden und damit gleichzeitig die Relevanz und Rolle der Mathematik für die Entwicklung von KI-Systemen herausgestellt werden. Das Lernmaterial soll zur Entmystifizierung von KI beitragen, indem es aufzeigt, wie solche Systeme mithilfe von Mathematik entwickelt werden und welche Rolle der Mensch bei ihrer Entwicklung spielt. Dabei sollen die Lernenden ausgewählte mathematische Methoden, die im Bereich des NLP verwendet werden, kennen lernen und problembezogen anwenden. Um insbesondere die Bedeutung der Schulmathematik für die Entwicklung von KI-Systemen herauszustellen, soll an das jeweilige Vorwissen angeknüpft werden. Die mathematischen Methoden können aber (notwendigerweise) zum Teil auch über das Niveau des Schulstoffes

hinausgehen, weshalb das Lernmaterial einen Ausblick auf hochschulmathematische Inhalte gibt. Bei der Erarbeitung neuer Inhalte insbesondere neuer komplexer Inhalte kann das Material lediglich als Einführung dienen. Für eine nachhaltige Festigung neuer mathematischer Inhalte sind vertiefende Übungen und weitere Anwendung der mathematischen Methoden auf andere Kontexte notwendig.

Der Schwerpunkt liegt weniger auf dem Ausbau von Modellierungskompetenzen, sondern vielmehr darauf, die Bedeutung der Mathematik im Alltag der Lernenden zu betonen. Ziel ist es, die Einstellung gegenüber dem Fach Mathematik positiv zu beeinflussen, indem deren Relevanz für praxisnahe und moderne Themen wie KI hervorgehoben wird.

Auf der anderen Seite soll das Lernmaterial, indem es über die Funktionsweise von KI-Systemen und deren mathematische Grundlage aufklärt, zu einem vertieften Verständnis von KI beitragen. Ziel ist es, Kompetenzen zu fördern, die für ein vertieftes Verständnis von KI essenziell sind. Hierzu werden in Kapitel 6 übergreifende Lernziele vorgestellt, welche als Orientierung bei der Entwicklung dienen.

Konkret wird exemplarisch Lernmaterial zu zwei Modellierungsprojekten mit unterschiedlichen Problemstellungen aus dem Bereich NLP entwickelt. Die erste Problemstellung befasst sich mit der Funktionsweise eines Assistenzsystems, das beim Verfassen von Nachrichten unterstützt, indem es Vorschläge für das nächste Wort generiert. Die zweite Problemstellung konzentriert sich auf die Erkennung gesprochener Sprache. Somit liegt der Fokus einmal auf der Entwicklung eines Sprachmodells zur Generierung von (textueller) Sprache und einmal auf der Entwicklung eines Modells mithilfe dessen (verbale) Sprache identifiziert wird. Diese Auswahl soll die Vielseitigkeit des Forschungsfeldes der Verarbeitung natürlicher Sprache verdeutlichen, indem sowohl die Aufgabenvielfalt mit den Schwerpunkten Sprachverstehen und Spracherzeugung als auch die verschiedenen Arten von Sprache, die von KI-Systemen verarbeitet werden können, thematisiert werden. Zudem resultiert die Auswahl der Problemstellungen aus der Berücksichtigung gewisser Kriterien, welche in Kapitel 7 bei der Formulierung der Design-Prinzipien für die Entwicklung von realitätsbezogenem Lernmaterial zum Thema KI dargelegt und schließlich in Abschnitt 14.2 und Abschnitt 20.2 in Bezug zu den ausgewählten Problemstellungen reflektiert werden.

Die Entwicklung des Materials basiert auf einer stoffdidaktischen Analyse der beiden Problemstellungen sowie der für deren Lösung erforderlichen mathematischen Inhalte. Angesichts der Vielzahl an mathematischen Inhalten, die für die Lösung der beiden Probleme notwendig sind, wird hier nur eine mögliche Lösung betrachtet und deren Zugänglichkeit untersucht. An geeigneten Stellen werden jedoch alternative Ansätze aufgezeigt und begründet, warum sie nicht weiterverfolgt werden oder inwiefern es denkbar wäre, auch diese didaktisch aufzubereiten. Dazu wird das Problem zunächst mit den ausgewählten Methoden mathematisch modelliert, um die relevanten mathematischen Inhalte identifizieren zu können. Auf dieser Basis erfolgt dann eine didaktische Reduzierung der mathematischen Inhalte, die in der Entwicklung von Lernmaterial mündet. Durch wiederholte Erprobung des Materials und die daraus gewonnenen Erkenntnisse wird dieses gezielt verbessert.

4.2 Abgrenzung zu bestehenden Forschungsarbeiten und Lernumgebungen

In diesem Abschnitt wird ein Überblick über die Veröffentlichungen zur Verarbeitung natürlicher Sprache als Unterrichtsinhalt gegeben. Besonderes Augenmerk liegt dabei auf Forschungsarbeiten, die entweder ebenfalls einen mathematischen Fokus haben oder die gleichen Unterthemen, wie Spracherkennung und N-Gramm-Modelle, adressieren und somit in enger Verbindung zur vorliegenden Arbeit stehen. Zudem werden Lernumgebungen mit ähnlicher thematischer Ausrichtung vorgestellt, und es erfolgt eine Abgrenzung des in dieser Arbeit entwickelten Lernmaterials. Es sei darauf hingewiesen, dass trotz großer Sorgfalt keine Garantie für die Vollständigkeit dieser Zusammenstellung gegeben werden kann.

Touretzky und Gardner-McCune (2023) nehmen die Spracherkennung mit Lernenden allerdings aus der sprachwissenschaftlichen Perspektive in den Blick. Hier geht es mitunter auch um die Funktionsweise, diese wird allerdings eher oberflächlich angesprochen. Zum Beispiel wird die Variabilität der Ergebnisse als Indikator für ein zugrundeliegendes statistisches Modell thematisiert sowie die Notwendigkeit von Transformationen um zu einem verarbeitbaren Datenformat zu gelangen. Eine weitere Thematisierung der Funktionsweise erfolgt nicht. Stattdessen sind zentrale Elemente die unterschiedlichen Strukturebenen von Sprache, phonologisch als Geräusch, lexikalisch als Wörter, grammatisch als Syntax, semantisch als Wortbedeutung und kulturell.

Weiterhin gibt es eine Zusammenstellung der Aufgaben des Wettbewerbs NACLO (North American Computational Linguistics Olympiad), in dem Schülerinnen und Schüler im außerschulischen Rahmen linguistische Probleme lösen (Radev, 2013). Diese Problemstellungen bestehen hauptsächlich aus kleinen Aufgaben aus den Bereichen Computerlinguistik und Informatik, die mit logischen und analytischen Überlegungen gelöst werden können. Komplexere Probleme, die das Erlernen neuer Methoden erfordern würden, werden hingegen nicht behandelt. Ein Beispiel dafür ist, dass nur ein Teilaspekt der Spracherkennung behandelt wird, indem das Lesen von Spektrogrammen fokussiert wird.

In Oldenburg (2008) wird tatsächlich stärker die Funktionsweise des auch in dieser Arbeit vorgestellten N-Gramm-Modells in den Blick genommen. Das Ziel ist dabei nicht das Vorschlagen eines Wortes, sondern, das Generieren von von ihm so bezeichneten Zufallssätzen. Der Fokus liegt hier auf der Informatik und der Umsetzung der Methode im Code. Es wird das Uni-, Bi- und Trigramm-Modell vorgestellt. Der mathematische Hintergrund wird wenig bis gar nicht beleuchtet.

Dewan und Touretzky (2024) präsentieren ein Demonstrationstool zum N-Gramm-Modell, das über einen Webbrowser zugänglich ist und speziell für den Einsatz mit Lernenden entwickelt wurde. Mit diesem Tool können durch Auslesen von Trainingstexten mit wenigen Mausklicks neue Texte generiert werden. Allerdings fehlen in dieser Arbeit konkrete Vorschläge und Implikationen für den Einsatz im Unterricht, wie beispielsweise Aufgabenstellungen oder eine klare Vorgabe eines Lernpfades für die Schülerinnen und Schüler.

Weitere Lernumgebungen, die für das Erlernen des Verarbeitens natürlicher Sprache konzipiert sind (zum Beispiel Zhorai¹, eCraft2Learn², Teachable Machine³, Machine Learning for Kids⁴), thematisieren grundlegende Strategien zur Entwicklung von KI-Systemen, bleiben die Funktionsweise betreffend jedoch sehr oberflächlich. Die Verarbeitung natürlicher Sprache spiegelt sich lediglich in der Art der Trainings- und Testdaten wider, während spezifische Methoden aus dem Bereich NLP unberücksichtigt bleiben.

Zudem soll ein kurzer Überblick über die Veröffentlichungen zu ML als Lerninhalt im Mathematikunterricht gegeben werden. Dabei werden auch Forschungsarbeiten berücksichtigt, die andere Anwendungsgebiete der KI als die Sprachverarbeitung behandeln.

Es ist zunächst festzustellen, dass es bisher insgesamt wenige Veröffentlichungen zu diesem Lerninhalt aus mathematikdidaktischer Perspektive gibt. Wang und Johnson (2019) geben einen kurzen Überblick über ausgewählte KI-Konzepte, erläutern die erforderlichen Kompetenzen für deren Verständnis und untersuchen, wie diese in US-amerikanischen High-School-Curricula für Mathematik integriert werden. Konkrete Überlegungen zur Umsetzung fehlen jedoch. Umfangreichere Forschung zu KI als Lerninhalt wird im Projekt ProDaBi⁵ betrieben. Beispielhaft seien hier die Arbeiten von Biehler und Fleischer (2021) sowie Opel et al. (2019) genannt, die digitale Lernmaterialien zu den ML-Methoden Neuronalen Netze und Entscheidungsbäume vorstellen. Der Fokus der entwickelten Lernarrangements liegt auf der Förderung von Kompetenzen aus der Statistik und der Informatik.

Weitere Forschungsarbeiten, in denen Entscheidungsbäume aus mathematikdidaktischer Perspektive analysiert werden, sind die Arbeit von Jablonka und Lengnink (2022), die sich auf die Förderung algorithmischer Mündigkeit konzentrieren und die Arbeit von Martignon et al. (2022), die den Fokus auf die Förderung von Risikokompetenzen legen. Bauer und Lengnink (2024) stellen Lernmaterial vor, in welchem Vorhersagen zur Vermittelbarkeit auf dem Arbeitsmarkt mit der Methode der logistischen Regression getroffen werden. Eine weitere relevante Veröffentlichung stammt von Hazzan und Mike (2022), die Lernmaterial mit Fokus auf der Statistik entwickelt haben. Sie untersuchen das k -nächste-Nachbarn-Verfahren hinsichtlich seiner Eignung für Data-Science-Kurse an Schulen aus pädagogischer und kognitiver sowie algorithmisch-didaktischer Perspektive und bereiten das Thema für den Einsatz in diesen Kursen auf.

Als am ähnlichsten zu dieser Forschungsarbeit erweisen sich die Arbeiten, welche ebenfalls im Projekt CAMMP entstanden sind. Diese verfolgen einen problembezogenen Ansatz, was bedeutet, dass Strategien und Methoden des ML und andere mathematische Inhalte ausgehend von einem Problem aus der Realität eingeführt werden. Das Lernmaterial unterscheidet sich im Anwendungskontext und teilweise

¹siehe zhorai.csail.mit.edu

²siehe <https://ecraft2learn.github.io/uui/index.html>

³siehe <https://teachablemachine.withgoogle.com/>

⁴siehe <https://machinelearningforkids.co.uk/>

⁵siehe www.prodabi.de

auch in der verwendeten ML-Methode von dem im Rahmen dieser Arbeit entwickelten Lernmaterial. So befassen sich Schönbrodt (2022) mit Support Vector Machines, die zur Entwicklung eines Bilderkennungssystems und eines Empfehlungssystems genutzt werden. Kindler et al. (2023) entwickeln Lernmaterial zu Neuronalen Netzen im Kontext der Vorhersage von Lebenserwartungen. In der Veröffentlichung von Schönbrodt et al. (2024) wird KI-Bildung in den Mathematikunterricht integriert, indem mit Lernenden das k -nächste-Nachbarn-Verfahren am Beispiel der Aktivitätserkennung erarbeitet wird.

Insgesamt widmet sich jedoch keine der genannten Forschungsarbeiten mit dem spezifischen Teilgebiet der KI - der Verarbeitung natürlicher Sprache - als Lerninhalt aus mathematikdidaktischer Perspektive. Das in Abschnitt 4.1 vorgestellte Forschungsvorhaben dieser Arbeit adressiert somit einen bisher wenig beachteten Bereich. Insbesondere die konkreten Anwendungen wie Spracherkennung oder Wortvorhersage bleiben unbeachtet. Das in dieser Arbeit behandelte N-Gramm-Modell, das zur Wortvorhersage genutzt wird, stellt eine bislang nicht mathematikdidaktisch untersuchte Methode dar. Auch in der allgemeinen Didaktikforschung finden sich nur wenige relevante Arbeiten zu NLP als Lerninhalt, und wenn NLP-Themen behandelt werden, bleibt die Analyse der Funktionsweise solcher Systeme meist oberflächlich.

Teil II

Lernziele, Design-Prinzipien und Rahmenbedingungen der computergestützten Modellierungsprojekte

5 Einleitung

Dieser Teil der Arbeit stellt die übergreifenden Lernziele, die Design-Prinzipien und Rahmenbedingungen der in dieser Arbeit entwickelten Modellierungsprojekte vor. Zentral sind dabei die Zielsetzungen des übergeordneten Programms *Computational and Mathematical Modeling Program*¹ (CAMMP), in dessen Kontext die Lehr- und Lernmaterialien entstehen.

CAMMP ist ein Schülerlabor an der Rheinisch-Westfälischen Technischen Hochschule Aachen sowie eine Initiative an der Paris Lodron Universität Salzburg sowie am Karlsruher Institut für Technologie als Teil des Projekts *Simulierte Welten*². Seit 2011 bietet es Modellierungsprojekte für Schülerinnen und Schüler der Jahrgangsstufen 7 bis 13 an. Das Ziel des Programmes ist es, die gesellschaftliche Bedeutung von Mathematik einer breiten Öffentlichkeit bewusst zu machen. Dazu soll durch die Entwicklung innovativer Konzepte sowie Lehr- und Lernmaterialien Schülerinnen und Schülern ermöglicht werden, möglichst aktiv in das Lösen außermathematischer, realer, relevanter, authentischer Probleme mithilfe mathematischer Modellierung und Computereinsatz einzusteigen. Die Mathematik wird im Rahmen von CAMMP-Projekten somit nie um der Mathematik willen betrieben, sondern stets zur Lösung dieser Probleme eingesetzt. Der Computer spielt dabei eine zentrale Rolle, doch sind die Materialien so konzipiert, dass keine Programmierkenntnisse erforderlich sind, um das Angebot allen Schülerinnen und Schülern zugänglich zu machen (Frank et al., 2022, S. 2).

Die Zielsetzungen des Programms CAMMP sind für die Definition übergreifender Lernziele und die Festlegung von Design-Prinzipien zentral. Kapitel 6 erläutert die übergreifenden Lernziele, die die angestrebten Kompetenzen beschreiben, die durch die Teilnahme an den Modellierungsprojekten gefördert werden sollen. Ihre Formulierung orientiert sich neben den im Projektrahmen von CAMMP vorgegebenen Zielen an einem in der Fachliteratur entwickelten Kompetenzrahmen für KI-Kompetenzen. Um die Erreichung der Lernziele mit einer geeigneten Gestaltung der Materialien zu unterstützen, werden in Kapitel 7 grundlegende Design-Prinzipien formuliert. Sie dienen als Orientierung für die Gestaltung der Lehr-Lernarrangements.

Darüber hinaus wird in Kapitel 8 und 9 der ganzheitliche Ansatz zur Durchführung kollaborativer Modellierungsaktivitäten vorgestellt, der im Rahmen des Programmes CAMMP entwickelt wurde. Dazu zählen der grundlegende Aufbau der Lernmaterialien für die Modellierungsprojekte (vgl. Kapitel 8) und der Ablauf der Modellierungsprojekte (vgl. Kapitel 9). Die in dieser Arbeit entwickelten Lernmaterialien werden

¹siehe www.cammp.online

²siehe simulierte-welten.de

dabei in zwei verschiedenen Modellierungsformaten erprobt: a) in Modellierungstagen, die sich über einen Tag erstrecken, und b) in Modellierungs-Doppelstunden mit einem kurzen Zeitrahmen von 1,5 Zeitstunden.

6 Übergreifende Lernziele

In Geisler et al. ([under review](#)) formuliert die Autorin übergreifende Lernziele für Modellierungstage mit KI-Inhalten, die als Orientierung für die Entwicklung der Lernmaterialien dienen. Die Formulierung dieser Lernziele basiert sowohl auf den durch den Projektrahmen des Schülerprojektes CAMMP vorgegebenen Zielsetzungen als auch auf in der Literatur definierten KI-Kompetenzen. Im Folgenden werden diese Lernziele vorgestellt und ihre theoretische Fundierung erläutert.

Aus Sicht der Mathematikdidaktik existiert bislang kein spezifischer Kompetenzrahmen für KI. Die Lernziele orientieren sich daher an den allgemeiner formulierten Kompetenzen, welche von Long und Magerko ([2020](#)) zu einer *AI Literacy* zusammengefasst wurden. Diese Kompetenzen sollen Lernende dazu befähigen

„[...] KI-Technologien kritisch zu bewerten; effektiv mit KI zu kommunizieren und zusammenzuarbeiten; und KI als Tool online, zu Hause und am Arbeitsplatz zu nutzen.“¹

[Übers. d. Verf.] (Long & Magerko, [2020](#), S. 598)

Der Kompetenzrahmen von Long und Magerko ([2020](#)), welcher in die Kategorien „What is AI?“, „What can AI do?“, „How does AI work?“, „How should AI be used?“ und „How do people perceive AI?“ unterteilt ist (Long & Magerko, [2020](#), S. 598), bietet aus mehreren Gründen eine geeignete Grundlage für die Formulierung der Lernziele. Zum einen wird er in der wissenschaftlichen Literatur als eine der einflussreichsten Arbeiten im Bereich der AI Literacy angesehen, was seine Relevanz und Akzeptanz unterstreicht (Casal-Otero et al., [2023](#), S. 5). Zum anderen bezieht er sich, ebenso wie die im Rahmen des Projektes CAMMP und somit auch die hier entwickelten Modellierungsprojekte, auf Lernende ohne technischen Hintergrund (Long & Magerko, [2020](#), S. 598). Zudem werden Programmierfähigkeiten – im Gegensatz zu anderen Veröffentlichungen – nicht als notwendige Kompetenz der AI Literacy angesehen (Laupichler et al., [2022](#), S. 13). Dies entspricht den Zielsetzungen des Projekts CAMMP, bei dem die hier zu konzipierenden Modellierungsprojekte den Fokus auf mathematische statt auf informatische Kompetenzen legen. Darüber hinaus legt der Rahmen besonderen Wert auf die Vermittlung der Funktionsweise von KI-Systemen und sieht das Maschinellen Lernen als zentralen Bestandteil von AI Literacy (Long & Magerko, [2020](#), S. 599 - 601).

Für die Formulierung der Lernziele werden diejenigen Kompetenzen von Long und Magerko ([2020](#)) berücksichtigt, welche die Funktionsweise von KI-Systemen betreffen. Dazu zählen Kompetenzen zur Repräsentation und Verarbeitung von Daten

¹ „to critically evaluate AI technologies; communicate and collaborate effectively with AI; and use AI as a tool online, at home, and in the workplace.“

(Representations, Data literacy), zum Lernen aus Daten (Learning from Data), zu den einzelnen Schritten des Maschinellen Lernens (ML-Steps) sowie zur Rolle des Menschen bei der Entwicklung von KI-Systemen (Human Role in AI) aus der Kategorie „How does AI work?“. Diese Kompetenzen werden aus einer mathematischen Perspektive spezifiziert und beinhalten sowohl konkrete mathematische Inhalte als auch grundlegende mathematische Prozesse. Die ausformulierten Lernziele lassen sich den einzelnen Schritten des Maschinellen Lernens (Datenaufbereitung, Modellauswahl, Trainingsphase, Testphase) zuordnen (vgl. Tabelle 6.1).

Zudem werden ethische Implikationen thematisiert, um den Lernenden eine reflektierte Auseinandersetzung mit den Chancen und Risiken der Entwicklung und Anwendung von KI-Systemen zu ermöglichen (AI Ethics). Durch die Behandlung ethischer Fragestellungen soll ermöglicht werden, sich eine fundierte Meinung über Nutzen und Gefahren von KI zu bilden. Ein solides Verständnis der Funktionsweise von KI-Systemen ist dabei essenziell, um ethische Herausforderungen – wie etwa Diskriminierung durch statistische Verzerrungen – überhaupt erkennen zu können.

Die übergreifenden Lernziele legen die angestrebten Kompetenzen fest, die durch die Teilnahme an den Modellierungsprojekten gefördert werden sollen und sind daher für die Entwicklung der Lernmaterialien leitend. Sie werden in Abschnitt 13.1 und Abschnitt 19.1 für das Lehr-Lernarrangement der jeweiligen Problemstellung präzisiert und in spezifische Teilziele übersetzt.

Tabelle 6.1: Übersicht über die Lernziele in den Modellierungstagen, aufgeteilt auf zentrale Schritte des Maschinellen Lernens, angelehnt an Geisler et al. (under review)

Schritte des ML	Lernziele	Zugehörige Kompetenz der AI Literacy von Long und Magerko (2020)
	Die Lernenden können ...	
Datenaufbereitung	<p>(L1) ... mathematische Repräsentationen von Daten sowie deren computergestützte Verarbeitung beschreiben.</p> <p>(L2) ... Daten explorieren und, falls notwendig, Daten aufbereiten und geeignet darstellen.</p>	Representations Data Literacy
Modellauswahl	<p>(L3) ... getroffene Annahmen erläutern und die gewählte Funktionenklasse benennen.</p> <p>(L4) ... die Rolle von Trainingsdaten im ML-Prozess erläutern.</p> <p>(L5) ... die Funktionsweise der gewählten ML-Methode beschreiben.</p>	ML-Steps (model selection), Human Role in AI
Trainingsphase	<p>(L6) ... falls bei der gewählten Modellklasse notwendig, ein Optimierungsproblem formulieren und die Parameter der gewählten Funktion als Lösung des Optimierungsproblems bestimmen.</p> <p>(L7) ... Überlegungen zur Verbesserung der Modellierung anstellen.</p> <p>(L8) ... die Validierung der ML-Methode und die Rolle, die Testdaten dabei spielen, erläutern.</p> <p>(L9) ... die aus dem entwickelten Modell gewonnene (mathematische) Lösung bewerten.</p>	Learning from Data, ML-Steps (train)
Testphase	<p>(L10) ... ausgewählte statistische Qualitätsmaße (bspw. Genauigkeit) zur Bewertung des Modells nennen und berechnen.</p> <p>(L11) ... Ergebnisse von statistischen Qualitätsmaßen hinsichtlich der Eignung des Modells für die Anwendung interpretieren.</p>	ML-Steps (train), Human Role in AI
KI-Ethik	(L12) ... zentrale ethische Aspekte hinsichtlich Entwicklung und Einsatz des KI-Systems nennen und beurteilen (z. B. Privatsphäre, Datenschutz, Diskriminierung).	ML-Steps (test)
		Data literacy
		AI Ethics

7 Design-Prinzipien zur Erstellung der Lehr- und Lernmaterialien

Ebenso bedeutend wie die Lernziele, welche übergreifend für die Lernmaterialien festgelegt werden, ist die Gestaltung der Materialien zur Erreichung dieser Ziele. Daher werden für die Erstellung adäquater Lehr- und Lernmaterialien und deren Einsatz übergreifende didaktische sowie didaktisch-methodische Design-Prinzipien formuliert (Prediger et al., 2012, S. 454). Sie geben den Rahmen dafür vor, wie die Inhalte vermittelt werden und sind bei der Entwicklung der Lehr- und Lernmaterialien leitend.

Die Formulierung der Design-Prinzipien berücksichtigt die zeitlichen Rahmenbedingungen sowie die Zielsetzungen des Lernarrangements. Zentrales Anliegen ist es, die Relevanz der Mathematik für den Alltag der Lernenden herauszustellen, indem Probleme aus der realen Welt betrachtet und mithilfe von Mathematik gelöst werden. Im Lernarrangement steht daher die mathematische Modellierung einer realen Situation im Zentrum. Um die Relevanz der Mathematik deutlich zu machen ergeben sich die Notwendigkeit, die Rolle der Mathematik herauszuarbeiten (DP4), sowie Anforderungen an die Art der Problemstellung (DP5). Darüber hinaus liegt ein besonderes Augenmerk auf der inner- und außermathematischen Interdisziplinarität der Problemstellung und ihrer Lösung (DP3).

Gleichzeitig soll ein Verständnis für die Funktionsweise von KI-Systemen aufgebaut werden, sodass die Lernenden Möglichkeiten, Grenzen und Chancen von KI-Systemen einschätzen können. Grundlegend dabei ist die Vermittlung von Wissen zu Konzepten und Methoden aus dem Maschinellen Lernen. Aufgrund der Komplexität und Fülle des zu erlernenden Wissens wird ein eher angeleiteter Lernansatz gewählt. Trotz des global stärker angeleiteten Ansatzes sollen jedoch Möglichkeiten zur Diskussion und Kommunikation über die Lerninhalte geschaffen werden (DP2). Außerdem werden Differenzierungsmaßnahmen (DP6) vorgesehen, um den unterschiedlichen Niveaus einer heterogenen Lerngruppe gerecht zu werden. Die Nutzung geeigneter technischer Hilfsmittel (DP1) ist ebenfalls essenziell, da sie die Bearbeitung datenintensiver Problemstellungen aus dem Kontext des Maschinellen Lernens erheblich erleichtert.

Da der Lerngegenstand Maschinelles Lernen im Kontext der Sprachverarbeitung noch sehr neu ist, ist bisher wenig fachdidaktisches Wissen dazu vorhanden. Um sich trotzdem an bewährten Kriterien orientieren zu können, werden Hintergrundtheorien zum Lehren und Lernen des mathematischen Modellierens und der Statistik berücksichtigt.

Da Maschinelles Lernen und Statistik inhaltlich eng verbunden sind, bieten sich Ideen für die Gestaltung von Statistik-Lernumgebungen als wertvolle Grundlage für das Design der hier zu entwickelnden Lernumgebung an. Die von Ben-Zvi et al. (2018) entwickelten Prinzipien für das Design einer „Statistical Reasoning Learning Environment“ haben sich im Zusammenhang mit dem Lehren und Lernen im Bereich Daten etabliert und dienen daher als Basis für die hier erstellten Design-Prinzipien. Die grundlegenden Ideen von Ben-Zvi et al. (2018) fließen insbesondere in DP1 und DP2 ein und unterstützen zudem DP5.

Im Kontext der Modellierung können Kriterien für gute Modellierungsaufgaben als leitende Prinzipien angesehen werden. Diese Kriterien sollen helfen, eine Auswahl geeigneter Problemstellungen für den mathematischen Modellierungsunterricht zu treffen. Etabliert sind insbesondere die von Pohjolainen und Heiliö (2016, S. 4) benannten Kriterien, die unter anderem bei anderen Veröffentlichungen aus dem Projekt CAMMP aufgegriffen werden (Frank et al., 2022, S. 4), weshalb diese bei der Formulierung der Design-Prinzipien von besonderer Wichtigkeit sind. Weitere Hintergrundliteratur aus der Modellierungsdidaktik, die zum Teil ebenfalls bereits in Frank et al. (2022, S. 2) aufgegriffen wird, wird als Untermauerung der Design-Entscheidungen herangezogen. Die Hintergrundtheorien zum Modellieren spiegeln sich in DP2, DP3, DP4 und DP5 wider.

Insgesamt ergeben sich sechs Design-Prinzipien, die im Folgenden zusammen mit ihren bereits angedachten Umsetzungsstrategien als Rahmen für die Entwicklung der Lehr- und Lernmaterialien dargelegt werden. Die Reihenfolge der Design-Prinzipien ist hierbei nicht als Gewichtung ihrer Bedeutung zu sehen:

Geeignete technische Hilfsmittel verwenden (DP1):

Die Nutzung technischer Hilfsmittel hat einen starken Einfluss auf die Art und Weise, wie Lernende auf Daten zugreifen und diese untersuchen können (Ben-Zvi et al., 2018, S. 5). Um ein Verständnis für die Prozesse des Maschinellen Lernens und insbesondere für die Rolle, die Daten darin innehaben, zu vermitteln, muss die digitale Lernumgebung für das Lösen datenlastiger Probleme geeignet sein. Das der Lernumgebung zugrunde liegende digitale Werkzeug muss in der Lage sein, große Datenmengen verarbeiten und auf verschiedene Art darstellen zu können. Die digitale Lernumgebung soll zudem so zugänglich gemacht werden, dass Schülerinnen und Schüler auch mit wenig oder keiner Programmiererfahrung in der Lernumgebung arbeiten können. Die digitale Lernumgebung soll die Möglichkeit bieten, den Lernprozess auf verschiedenen Ebenen zu unterstützen, sei es in Form von Entlastung beim Rechnen, mithilfe unterschiedlicher Darstellungsformen, verschiedener Differenzierungsmöglichkeiten oder eines Feedbacksystems. Eine ausführliche Erläuterung, inwiefern digitale Medien beim Modellieren unterstützen können, ist in Abschnitt 2.3 nachzulesen.

Diskussions- und Reflexionsanlässe in kooperativen Lernformen bieten (DP2):

Angesichts der erwarteten Komplexität und Vielfalt der mathematischen Methoden, die zur Entwicklung eines Systems zur Sprachverarbeitung erforderlich sind, wird

ein eher angeleiteter Ansatz bei der Gestaltung des Lernmaterials gewählt, um eine kognitive Überforderung der Lernenden zu vermeiden. Trotz dieses angeleiteten Ansatzes sollen Gelegenheiten für Reflexion und Diskussion über den Lerninhalt insbesondere zwischen den Lernenden integriert werden. Um diesen Aspekt zu berücksichtigen, soll die Bearbeitung des Materials in Partnerarbeit erfolgen. Diese Sozialform soll zugleich eine gegenseitige Unterstützung bei der Bearbeitung der Aufgaben ermöglichen und die Zusammenarbeit fördern. Dabei kann den Lernenden verdeutlicht werden, dass Teamarbeit bei der Bewältigung realer, komplexer Probleme von wichtiger Bedeutung ist (Pohjolainen & Heiliö, 2016, S. 4). Zudem sollen im Lernmaterial gezielt Aufgaben integriert werden, die den Austausch zwischen den Lernenden und Diskussionen über den Lerngegenstand erfordern. Dazu gehören Aufgaben zur Sammlung von Ideen für einzelne Teilschritte der Modellentwicklung und von Ideen für Modellverbesserungen sowie Reflexionsfragen. Die Förderung von Kommunikationskompetenzen soll außerdem durch Plenumsdiskussionen gewährleistet werden, in denen Teilergebnisse gesichert, offene Fragen erörtert und Ideen der Lernenden diskutiert werden. Insgesamt soll das Lernarrangement dazu anregen, Lösungsansätze zu erläutern und zu begründen, Erklärungen und Argumente von anderen Lernenden aufmerksam nachzuvollziehen und alternative Denkansätze kritisch zu hinterfragen (Ben-Zvi et al., 2018, S. 7).

Innermathematische und außermathematische Interdisziplinarität integrieren (DP3):

Die Modellierung realer Probleme bietet Gelegenheit, sowohl außer- als auch innermathematische Beziehungen aufzuzeigen. Dieses Potential soll im Lernarrangement ausgenutzt werden, um ein vernetzendes Lernen zu ermöglichen. Für Pohjolainen und Heiliö (2016, S. 4) zählt die Verknüpfung verschiedener mathematischer Inhalte zu den Kriterien für gute Modellierungsprobleme. Durch die Schwerpunktsetzung dieses Dissertationsprojektes auf die Vermittlung mathematischer Methoden, die für die Verarbeitung natürlicher Sprache eine wichtige Rolle spielen, greifen die entwickelten Modelle natürlicherweise Inhalte aus ganz unterschiedlichen Bereichen der Mathematik (bspw. Geometrie, Analysis, Stochastik, lineare Algebra) auf und verknüpfen diese. Die innermathematische Vernetzung kann im Lernmaterial daher leicht hervorgehoben werden. Auf diese Weise sollen tragfähige Sinnzusammenhänge aufgebaut werden.

Neben dem Hervorheben der innermathematischen Beziehungen liegt ein zentrales Element der Sinnkonstruktion im Mathematikunterricht im Aufzeigen der Beziehung der mathematischen Begriffe zur Umwelt der Lernenden und damit in der Betonung der außermathematischen Interdisziplinarität der Problemstellung. Freudenthal (1973, S. 77) bezeichnet dies als das Prinzip der Beziehungshaltigkeit und schreibt dazu

„Will man zusammenhängende Mathematik unterrichten, so muss man in erster Linie die Zusammenhänge nicht direkt suchen; man muss sie längs der Ansatzpunkt verstehen, wo die Mathematik mit der erlebten Wirklichkeit des Lernenden verknüpft ist. Das – ich meine die Wirklichkeit – ist das Skelett, an das die Mathematik sich festsetzt.“
(Freudenthal, 1973, S. 77).

Das Prinzip der Beziehungshaltigkeit steht in enger Beziehung zur Modellierung, bei der Probleme aus der Realität in mathematischer Form dargestellt und gelöst werden. Lernende erleben bei der Bearbeitung von Modellierungsaufgaben, wie Mathematik in der Umwelt angewendet werden kann. Für die Lösung der allermeisten realen Probleme ist es dabei notwendig, unterschiedliche Fachperspektiven zu integrieren. Beispielsweise erfordert die Lösung von Problemstellungen aus dem Bereich der Sprachverarbeitung neben der Nutzung mathematischer Methoden auch außermathematisches Wissen, primär aus der Linguistik und der Informatik aber zum Teil auch aus der Physik. Pohjolainen und Heiliö (2016, S. 4) bezeichnen dies als die „multidisziplinäre Natur der Modellierung“¹ [Übers. d. Verf.], welche ihrer Meinung nach in Lernmaterialien zur Modellierung beleuchtet werden soll. Sie setzen die Multidisziplinarität in direkten Zusammenhang mit der Rolle der Teamarbeit beim Lösen realer Probleme, die in der Wirtschaft und Forschung eine automatische Folgerung der Interdisziplinarität des Problems ist (vgl. DP2). Forschende aus unterschiedlichen Fachdisziplinen arbeiten parallel und zumeist auch gemeinsam an der Entwicklung eines Modells. Da die mathematischen Inhalte jedoch im Zentrum stehen sollen, soll das neben dem Alltagswissen benötigte außermathematische Wissen im Lernmaterial in Form von ausführlichen Erläuterungen bereit gestellt werden (Frank et al., 2022, S. 2).

Rolle der Mathematik herausarbeiten (DP4):

Dass besonders die Mathematik für die Entwicklung von Systemen der Sprachverarbeitung relevant ist, soll deutlich gemacht werden, indem im Lernmaterial die zentralen mathematischen Inhalte bei der Modellentwicklung und -validierung herausgearbeitet werden. Um den Nutzen mathematischer Modelle zu verdeutlichen (Pohjolainen & Heiliö, 2016, S. 4), soll mit den Lernenden im Modellierungsprojekt ein didaktisch reduziertes aber dennoch funktionsfähiges KI-System entwickelt werden. Dabei wird, wie auch von Ben-Zvi et al. (2018, S. 3) empfohlen, der Fokus auf die zentralen mathematischen Inhalte gelegt. Im Gegensatz zur Auffassung von Ben-Zvi et al. (2018, S. 3) wird jedoch nicht davon ausgegangen, dass die Lernenden alle mathematischen Ideen selbst entwickeln, da diese bei der Konstruktion eines KI-Systems zahlreich und teilweise komplex sind. Neue mathematische Inhalte werden dann im Gegensatz zu einem methodenorientierten Ansatz problembezogen eingeführt. Dies bedeutet, dass mathematische Strategien und Methoden erst dann erlernt werden, wenn sie für die Modellierung des Problems vonnöten sind. In diesem Sinne wird den Lernenden Gelegenheit gegeben,

„Erscheinungen der Welt um uns, die uns alle angehen oder angehen sollten, aus Natur, Gesellschaft und Kultur, in einer spezifischen Art wahrzunehmen und zu verstehen [...]“
(Winter, 1995, S. 37).

Dies stellt die erste der drei Grunderfahrungen dar, die laut Winter (1995) im Mathematikunterricht angestrebt werden sollten (vgl. Abschnitt 2.2).

Das Lernmaterial soll daher ermöglichen, den gesamten Modellierungsprozess vollständig und mehrfach zu durchlaufen. Um auch komplexere reale Probleme in einem

¹„multidisciplinary nature of modelling“

begrenzten Zeit- und Betreuungsrahmen mit den Lernenden lösen zu können, wird ein Ansatz verfolgt, der zwischen dem atomistischen Modellieren (Blomhoej & Højgaard, 2003, S. 128 - 129) und dem holistischen Modellieren (Blomhoej & Højgaard, 2003, S. 128) anzusiedeln ist. Dabei werden alle Modellierungsschritte, wenn auch abgegrenzt voneinander, durchlaufen. Die Unterteilung des Modellierungsprozesses in kleinere, angeleitete Teilaufgaben soll helfen, schrittweise mathematische Formeln und Verfahren zu entwickeln und die Lösung von realen Problemen nachzuvollziehen. Durch die Möglichkeit der somit weitestgehend selbstgesteuerten Bearbeitung der Aufgaben sollen die Lernenden aktiviert werden.

Um das Verständnis für die zur Lösung des Problems angewandten mathematischen Methoden zu fördern, sollen die dafür relevanten Berechnungen exemplarisch anhand einfacher Beispiele durchgeführt werden. Hazzan und Mike (2023, S. 237) bezeichnen diese Methode als „Hands-On-Tasks“ und heben deren Bedeutung für das Verständnis von Methoden des Maschinellen Lernens heraus. Der Computer kann anschließend genutzt werden, um die erforderlichen Berechnungen repetitiv und eventuell leicht angepasst auf komplexere Ausgangssituationen auszuführen. Um die Hürden für die Lernenden weiterhin zu verringern und sich dennoch mit realen Problemen beschäftigen zu können, wird wie von Pohjolainen und Heiliö (2016, S. 4) vorgeschlagen, zunächst ein möglichst einfaches, leicht zugängliches Modell konstruiert. Dieses kann anschließend durch die immer detailliertere Anpassung der Annahmen an die Problemsituation sowie daraus resultierende Verbesserungen präzisiert werden. Das Lernarrangement soll jedoch auch aufzeigen, dass eine perfekte Lösung zumeist nicht erreichbar ist (Pohjolainen & Heiliö, 2016, S. 4), weil die Realität sich nicht komplett in einem Modell abbilden lässt und Vereinfachungen vorgenommen werden müssen, um die Realsituation handhabbar zu machen. Gleichzeitig können damit auch Grenzen der auf Basis der mathematischen Modelle entwickelten KI-Systeme herausgestellt werden.

Authentizität und Relevanz der Problemstellung berücksichtigen (DP5): Um die Bedeutung der Mathematik für reale Probleme ersichtlich zu machen, werden bei der Auswahl und Ausgestaltung der Problemstellung verschiedene Aspekte der Authentizität berücksichtigt. Authentizität meint dabei nach der Definition von Niss (1992, S. 353), dass die Problemstellung in reale Praktiken oder Fachgebiete außerhalb der Mathematik eingebettet ist und sich mit üblichen Problemen dieses Bereichs befasst. Dabei kann und soll das Modellierungsproblem insbesondere nicht nur hinsichtlich des Sachkontextes an sich authentisch sein, sondern auch hinsichtlich der Fragestellung und der Verwendung der mathematischen Methode (Palm, 2009, S. 3 - 19; Vos, 2011, S. 719 - 721). Authentizität im Bezug auf die Fragestellung bedeutet hierbei, dass diese von Forschenden aus dem entsprechenden eingebetteten Fachgebiet ebenfalls gestellt werden würde. Die Verwendung der für die Lösung des Problems genutzten Mathematik ist authentisch, wenn sie realistisch im Hinblick darauf ist, dass sie in der Realität (ggf. in komplexerer Form) tatsächlich zum Lösen der Probleme verwendet wird (Vos, 2011, S. 720 - 721). Insbesondere die Authentizität der Fragestellung wird von Vos (2018, S. 4) als wichtig für die Motivation und Leistung der Lernenden angesehen. Der Begriff der Authentizität lässt sich auf weitere Aspekte einer Aufgabenstellung übertragen (Vos, 2018, S. 12). Für diese Arbeit

soll weiterhin die Authentizität der verwendeten Daten berücksichtigt werden. Auch Ben-Zvi et al. (2018, S. 4) merken an, dass reale oder zumindest realistische Datensätze verwendet werden sollen. Insgesamt kann die Anwendung der Mathematik in authentischen Szenarien dazu beitragen, die Wichtigkeit der Mathematik zu zeigen.

Zudem sollte das betrachtete Modellierungsproblem relevant sein. Dabei wird der Begriff der Relevanz auf Burkhardt (1989, S. 9) zurückgeführt, welcher Modellierungsprobleme nach dem Interesse, den Lernende am Kontext haben können, klassifiziert. Problemkontakte, die für Schülerinnen und Schüler interessant sein könnten, werden darauf beziehend als relevant bezeichnet. Relevante Probleme haben Vorteile in Bezug auf Motivation, Engagement, Verständnis für die praktische Situation und die Fähigkeit, das Modell zu bewerten (Burkhardt, 1989, S. 9). Ob Lernende einen Kontext als relevant empfinden, wird allerdings nicht allein durch die Aufgabe bestimmt, sondern auch durch persönliche Interessen und Lebensumstände der einzelnen Lernenden und kann daher stark schwanken (Busse, 2005, S. 358). Es sollen deshalb Kontexte ausgewählt werden, von denen dabei ausgegangen werden kann, dass sie zumindest für den Großteil der Lernenden relevant sind. So kann die Relevanz der Mathematik für ihre jeweilige Lebenswelt erfahrbar gemacht werden, was als eines der übergeordneten Ziele des Lernmaterials festgelegt wurde.

Binnendifferenzierung ermöglichen (DP6):

Ein weiteres Prinzip, das aufgrund der Ausrichtung des Lernmaterials auf heterogene Lerngruppen unverzichtbar ist, betrifft die Binnendifferenzierung, auch als innere Differenzierung (Klafki & Stöcker, 1985) bekannt. Das Material soll die Möglichkeit bieten, innerhalb der Lerngruppe unterschiedliche Lernende individuell zu fördern, ohne die gesamte Gruppe dauerhaft aufzuteilen. Durch (gestufte) Hilfen zu einzelnen Aufgaben, vertiefendes Informationsmaterial und Zusatzaufgaben soll auch in heterogenen Gruppen die Möglichkeit geschaffen werden, eigenständig zu arbeiten. Um die eigenständige Arbeit zu unterstützen, sollen die Lösungen nach der Eingabe in digitale Arbeitsblätter automatisch überprüft werden. Dies soll ein selbstgesteuertes Arbeiten im eigenen Tempo ermöglichen, ohne die Lösungen direkt preiszugeben. Stattdessen sollen Schülerinnen und Schüler ermutigt werden, eigene Fehler zu finden und das bisherige Vorgehen zu überdenken und gegebenenfalls anzupassen.

Nur wenige der von Ben-Zvi et al. (2018) vorgeschlagenen Design-Prinzipien und der von Pohjolainen und Heiliö (2016, S. 4) formulierten Kriterien werden in den Design-Prinzipien dieses Projekts nicht umgesetzt. Eine Ausnahme bildet die Empfehlung, verschiedene Problemstellungen zu betrachten (Pohjolainen & Heiliö, 2016, S. 4). Diese Empfehlung kann aufgrund der zeitlichen Begrenzung der Modellierungsprojekte nicht berücksichtigt werden. Um ein vertieftes Verständnis für KI-Systeme zu schaffen, ist es notwendig, inhaltlich tief in die mathematischen Methoden einzusteigen. Innerhalb eines einzigen Projekttages lässt sich dies nur an einem ausgewählten Beispiel realisieren, da die begrenzte Zeit nicht ausreicht, um die Funktionsweise verschiedener KI-Systeme und die dahinterliegenden mathematischen Konzepte zu analysieren.

Ein weiterer Punkt betrifft die Forderung nach der Durchführung von Assessments, wie von Ben-Zvi et al. (2018, S. 6) vorgeschlagen. Diese bleibt beim Design der Modellierungsprojekte unberücksichtigt. Hauptgründe hierfür sind die zeitlichen Einschränkungen, die wenig Raum für zusätzliche Leistungstests lassen, sowie die Tatsache, dass die Projekte bisher primär in außerschulischen Veranstaltungen durchgeführt werden. Dennoch könnte dies in zukünftigen Iterationen des Konzepts und zwar insbesondere dann, wenn die Materialien auch verstärkt im Regelunterricht eingesetzt werden sollen, überdacht und angepasst werden.

Die konkrete Umsetzung der Design-Prinzipien wird an mehreren Stellen dieser Arbeit, aber insbesondere in den Abschnitten 14.2 und 20.2, im Detail exemplifiziert.

8 Aufbau des Lehr- und Lernmaterials

8.1 Aufbau des digitalen Lernmaterials

Wie in Abschnitt 2.3 bereits dargelegt, können digitale Medien gerade bei der Lösung realer Probleme den Lernprozess der Schülerinnen und Schüler sinnvoll unterstützen. Das in dieser Arbeit vorgestellte Lernmaterial ist daher zu einem großen Teil in Form von digitalen Arbeitsblättern umgesetzt. Auf die digitalen Lernmaterialien kann über eine Cloud-Plattform des KIT zugegriffen werden. Das Material lässt sich damit direkt im Webbrowser bearbeiten, sodass keine Installation von Software notwendig ist. Für die Nutzung der Inhalte wird lediglich ein digitales Endgerät mit stabiler Internetverbindung benötigt, sowie für akustische Komponenten ein Lautsprecher und ein Mikrofon.

Die Lernplattform ist über workshops.cammp.online erreichbar. Nach Anlegen eines Accounts auf der Plattform kann über ein Dropdown-Menü auf der Willkommensseite ein Workshop ausgewählt und das zugehörige Lernmaterial in den Account geladen werden (vgl. Abbildung 8.1).¹ Das Lernmaterial liegt anschließend als Ordner in der linken Menüleiste vor. Im Unterordner „worksheets“ liegen die digitalen Arbeitsblätter, die sich mit einem Doppelklick öffnen und anschließend bearbeiten lassen. Alle haben somit einen persönlichen Account mit einem eigenen, bearbeitbaren Satz an Lernmaterialien. Der Arbeitsstand wird automatisch zwischengespeichert, sodass er nach Pausen oder bei der Fortführung der Modellierungsaktivität über mehrere Einheiten hinweg jederzeit wiederhergestellt werden kann. Die Bearbeitung der Lernmaterialien im Account kann nur durch die Lernenden selbst erfolgen. Andere Lernende können nicht auf fremdes bearbeitetes Material zugreifen und somit keine Änderungen daran vornehmen.

Das Lernmaterial besteht aus verschiedenen Elementen, deren Ausgestaltung aus den Design-Prinzipien (vgl. Kapitel 7) abgeleitet werden. Nachfolgend werden die einzelnen Elemente im Detail – beginnend mit den digitalen Arbeitsblättern als Herzstück des Lernmaterials – vorgestellt und in Bezug zu den jeweiligen Design-Prinzipien gesetzt. Die Design-Entscheidungen zu den einzelnen Elementen bauen

¹Für das Anlegen eines Accounts ist beim Usernamen darauf zu achten, dass dieser mit dem Präfix „cammp_“ beginnt. Das Passwort kann frei gewählt werden. Eine detaillierte Beschreibung des Anmeldeprozesses zur Plattform findet sich als Video unter www.cammp.online/214.php. Auf die Lernmaterialien mit Lösung und das zugehörige Begleitmaterial kann durch Auswählen der Rolle „Lehrkraft“ und Eingabe des Passwortes „CAMMP=MatheFun!“ auf der Willkommensseite zugegriffen werden.

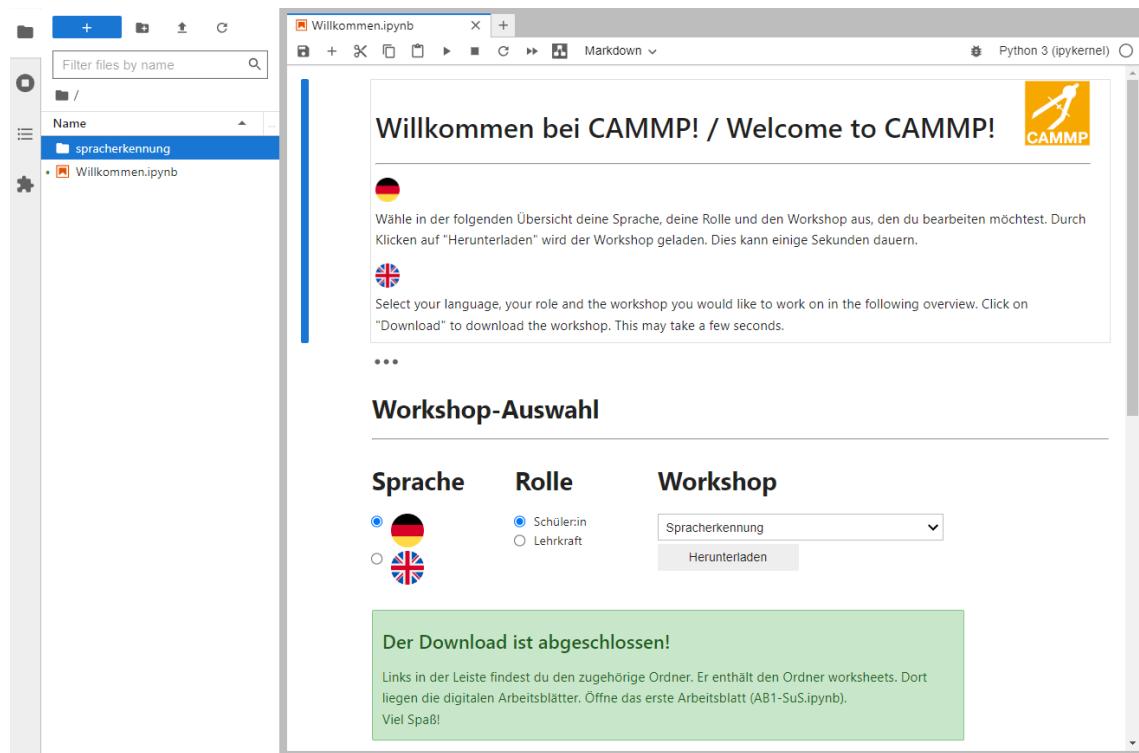


Abbildung 8.1: Bildschirmfoto der Willkommensseite der Lernplattform

auf den in Schönbrodt et al. (2021) beschriebenen früheren Arbeiten auf und erweitern diese.

8.1.1 Digitale Arbeitsblätter

Die digitalen Arbeitsblätter sind das Herzstück des Lernmaterials. Es handelt sich um sogenannte *Jupyter Notebooks*², beispielhaft dargestellt in Abbildung 8.2. Mit dem Begriff „Jupyter Notebook“ wird sowohl die web-basierte interaktive Entwicklungsumgebung bezeichnet, mit der Dokumente mit live-Code erstellt werden können, als auch das interaktive Dokument selbst. Wird hier von Jupyter Notebooks gesprochen, so ist stets das Dokument als Endprodukt gemeint.

Jupyter Notebooks gelten aufgrund ihrer breiten Anwendung in Industrie und Forschung (Pimentel et al., 2021, S. 1 - 2) als authentische Werkzeuge. Auch in der Forschung im Bereich der Verarbeitung natürlicher Sprache kommen sie häufig zum Einsatz, was die Vielzahl an Notebooks, die NLP-Bibliotheken nutzen, verdeutlicht (Neglectos, 2018). An den wenigen Stellen, an denen das Jupyter Notebook als Werkzeug in Schülerhand genutzt wird, kann den Lernenden somit ein Einblick in gängige Werkzeuge der MINT-Berufe gegeben werden. Zumeist werden die Jupyter Notebooks in den Modellierungsprojekten allerdings didaktisch aufbereitet und als digitale Lernumgebung eingesetzt. Für die Gestaltung als Lernumgebung eignen

²Für weitere Informationen siehe www.jupyter.org

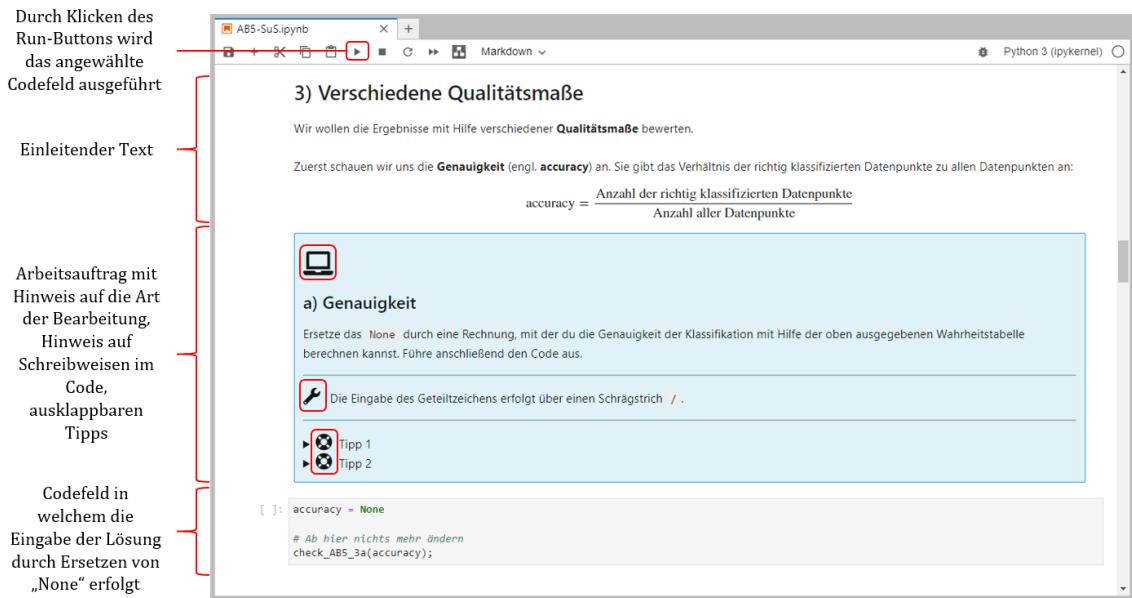


Abbildung 8.2: Bildschirmfoto eines digitalen Arbeitsblatts aus dem Lernmaterial zu Spracherkennung

sie sich besonders gut, weil sie viele verschiedene Bausteine übersichtlich in einer einzigen Datei beinhalten. Gegenüber reinen Code-Dateien bieten sie den Vorteil, dass Arbeitsaufträge, Formeln, Abbildungen, aber auch Codefelder direkt nebeneinanderstehen und somit das Arbeiten der Lernenden im Workshop erleichtern. Die Unterteilung des Problems in kleinere Aufgaben in einzelnen Abschnitten der Notebooks ebenso wie adaptive, automatisierte Rückmeldungen zu den Lösungen ermöglichen eine interaktive und sehr selbstgesteuerte Bearbeitung des Materials. Die Möglichkeit, weitere Notebooks mit Tipps oder Zusatzaufgaben unter den Arbeitsaufträgen zu verlinken, erlaubt eine Differenzierung des Lernprozesses. Insgesamt sind Jupyter Notebooks ein digitales Werkzeug, mit dessen Hilfe rechenlastige Prozesse ausgelagert und verschiedene Darstellungen von Daten einfach umgesetzt werden können. In geeigneter Gestaltung können die Notebooks als Lernumgebung fungieren, die den Lernprozess auf verschiedenen Ebenen unterstützen kann. Dadurch wird das erste Design-Prinzip im Modellierungsprojekt umgesetzt (vgl. Kapitel 7; DP1).

Die Gestaltung der Jupyter Notebooks erfolgt über die Bearbeitung zweier grundlegender Elemente, die sogenannten Text- und Codefelder. Die beiden Elementtypen können in flexibler Reihenfolge und mit beliebiger Häufigkeit im Notebook angeordnet werden. Im Folgenden werden die Text-, Codefelder und deren Ausgabe sowie deren Gestaltung im Lernmaterial näher beschrieben.

Textfelder

Die Textfelder bieten die Möglichkeit, erklärenden Text, Bilder, aber auch mathematische Gleichungen und Formeln darzustellen (in LaTeX-Notation). Des Weiteren gibt es die Möglichkeit, auf andere Dokumente, Arbeitsblätter oder Tipps über einen Link zu verweisen.

Tabelle 8.1: Symbole im Lernmaterial und deren Bedeutung

Symbol	Bedeutung
	Eingabe im Codefeld
	Textantwort oder zusätzliche Notizen erforderlich
	Diskussion mit Mitlernenden erforderlich
	Recherche notwendig
	Zeigt, welche Variablen und Schreibweisen für die Bearbeitung der Aufgabe wichtig sind.
!	Hinweis
	Tipp
	wichtige Information
	weiterführende Information
	Zeigt an, dass die gegebene Antwort korrekt ist.
	Zeigt an, dass die gegebene Antwort nicht korrekt ist.
	Lösung

Im Lernmaterial werden die Textfelder für informative Texte und Arbeitsaufträge genutzt. Jedes Arbeitsblatt beginnt und endet mit einem Textfeld, indem zu Beginn das Ziel des jeweiligen Arbeitsblatts benannt und zum Schluss der Inhalt des Arbeitsblatts in einem Satz zusammengefasst und ein Link zum nachfolgenden Arbeitsblatt bereitgestellt wird. Die weiteren Textfelder enthalten Arbeitsaufträge, Zwischeninformationen und Zwischenfolgerungen. Arbeitsaufträge sind dabei immer farblich hervorgehoben sowie durchnummeriert, sodass sie direkt gesehen und einfach wiedergefunden werden können. Ein kleines Symbol am oberen linken Rand der Aufgabenstellung zeigt an, welche Art der Bearbeitung erforderlich ist (vgl. Abbildung 8.2). Ein kleines Laptopsymbol weist auf die Bearbeitung direkt im Codefeld auf dem digitalen Lernmaterial hin, während das Stiftsymbol deutlich macht, dass diese Aufgabe auf dem Papier, oder, falls vorhanden, in einem Freitextfeld erfolgt. Eine Kombination aus Stift- und Laptopsymbol soll darauf hinweisen, dass die Antwort zwar im Codefeld erfolgt, Notizen für die Bearbeitung der Aufgabe aber sinnvoll sein können. Weitere Symbole in den Textfeldern werden verwendet, um die Texte zu strukturieren und auf besondere Punkte hinzuweisen. Eine Übersicht über die Symbole ist Tabelle 8.1 zu entnehmen.

Codefelder

Codefelder ermöglichen es, Code zu schreiben und den Code des einzelnen Codefeldes

auszuführen. Das Ausführen eines Codefeldes kann über den Run-Button (►) oder die Tastenkombination *Shift* und *Enter* erfolgen.

Im Lernmaterial wird ein Lückentext-Ansatz verfolgt. Das bedeutet, dass die Lernenden im Code die Lücken entsprechend der vorhergehenden Aufgabenstellung mit Formeln, Zahlen, Gleichungen oder (seltener) Begriffen ersetzen. Eine Lücke wird im Code durch den Platzhalter „NaN“ beziehungsweise „None“ gekennzeichnet. Kommentare im Code, die in Grau dargestellt und mit einer Raute gekennzeichnet sind, weisen zusätzlich die Stellen im Code aus, in welchen etwas ergänzt werden soll oder die nicht bearbeitet werden sollen (vgl. Abbildung 8.2). Durch diesen Ansatz sind keine Programmierkenntnisse notwendig. An wenigen Stellen werden einfache if-Anweisungen oder for-Schleifen im Code verwendet. Auch hier sind jedoch lediglich einzelne Elemente zu ergänzen, und die Struktur und ihre mathematische Funktion wird in ergänzenden Informationsblättern erläutert. Eigenständiges Programmieren wird lediglich auf manchen Zusatzblättern erforderlich. Dadurch bleibt der Fokus klar auf der Mathematik und der Modellierungsaufgabe, während gleichzeitig das algorithmische Denken gefördert wird, ohne Lernende aufgrund fehlender Programmierkenntnisse auszubremsen.

Das Lernmaterial zum Thema Wortvorhersage wurde in der Programmiersprache Julia und das später entstandene Material zur Spracherkennung in der Programmiersprache Python verfasst. Die Entscheidung für einen Wechsel zur Programmiersprache Python wurde im Rahmen des gesamten Projekts getroffen, wobei die Wahl auf Python aufgrund dessen Popularität und der Verfügbarkeit qualitativ hochwertiger Pakete, insbesondere für interaktive Codeelemente in Jupyter Notebooks, fiel. Im digitalen Lernmaterial zum Thema Spracherkennung sind daher besonders viele interaktive Elemente integriert, da deren Umsetzung in der gewählten Programmiersprache technisch einfacher realisierbar war. Die interaktiven Elemente werden im Lernmaterial als Ausgabe von Codefeldern erzeugt. Diejenigen Codefelder werden im Lernmaterial automatisch ausgeführt. Besonders häufig verwendete interaktive Elemente sind beispielsweise Tabellen, die befüllt werden können, Freitextfelder, Schieberegler, mit denen einzelne Komponenten eines Diagramms geändert werden können, Auswahlkästchen oder Audioplayer. Diese Elemente machen die Gestaltung von Ankreuzaufgaben oder auch Aufgaben, in denen etwas angehört werden soll, möglich. Zudem wurden Aufgaben implementiert, in denen Elemente sortiert werden müssen. Die neue Aufgabengestaltung macht die Lernumgebung interaktiver, passgenauer, und sie gibt den Lernenden die Möglichkeit, mehr selbstständig zu erkunden (zum Beispiel durch das Betätigen von Schieberegbern).

Ausgabe

Nach der Bearbeitung des Codefeldes führen die Lernenden die entsprechende Codezelle aus und erhalten direkt unter dem Codefeld eine Ausgabe. Diese enthält ein Feedback zur Korrektheit der Lösung und kann zusätzlich Bilder, Tabellen, Diagramme, Audiospuren oder einfache Ergebnisse mathematischer Berechnungen umfassen (vgl. Abbildung 8.3). Das Feedback für die Eingabe wird mit einer Überprüfungsfunktion generiert, die beim Ausführen der Codezelle aufgerufen wird. Der Code für die Überprüfungsfunktionen ist im Ordner „code“ ausgelagert und auf den digitalen Arbeitsblättern nicht sichtbar. Dadurch bleibt die Übersichtlichkeit des Ar-

beitsblatts erhalten, und es wird sichergestellt, dass keine ablenkenden Elemente im Lernmaterial enthalten sind.

Das Feedback enthält ein Symbol (Häkchen oder Kreuz; vgl. Tabelle 8.1), das zusätzlich zur farblichen Hervorhebung (grün für korrekt beziehungsweise rot für fehlerhaft) Aufschluss auf die Korrektheit der Lösung gibt. Bei Aufgaben, bei denen mögliche Fehlerursachen leicht erkennbar sind oder bestimmte Fehler häufiger auftreten, wird die Eingabe auf diese Fehler hin überprüft, und eine individuelle Rückmeldung wird im Text gegeben. Diese Rückmeldung enthält Denkanstöße, wie die Eingabe verbessert werden kann, oder weist darauf hin, wo im Code der Fehler passiert ist, wodurch die Fehlersuche erleichtert und das eigenständige Verbessern gefördert wird.

Manche Aufgaben enthalten kein automatisches Feedback. Dies kann zum einen daran liegen, dass die Lernenden in diesen Aufgaben ihre Lösungen eigenständig reflektieren sollen. In diesem Fall werden sie in der Aufgabe dazu aufgefordert, ihre Antworten beispielsweise durch die Betrachtung des ausgegebenen Diagramms oder der Audioausgabe selbst zu überprüfen. Durch das Ausprobieren unterschiedlicher Eingaben können sie den Zusammenhang zwischen Eingabe und Ergebnis genauer untersuchen und verstehen. Zum anderen ist ein automatisches Feedback nur bei geschlossenen Aufgabenformaten möglich. Lösungen zu Aufgaben, die nicht zu dieser Kategorie zählen, können in Einzelgesprächen mit der Lehrkraft oder im Plenumsgespräch besprochen werden. Dazu zählen insbesondere Aufgaben, in deren eine Interpretation der Ergebnisse gefordert wird oder offene sowie komplexere Zusatzaufgaben, deren Lösungswege nicht eindeutig sind oder einer ausführlicheren Erklärung bedürfen.

Neben inhaltlich-mathematischen Fehlern können auch syntaktische Fehler bei der Eingabe im Code gemacht werden. Dazu zählen beispielsweise falsch geschriebene mathematische Symbole oder ein inkorrekt Zugriff auf implementierte Befehle. Diese syntaktischen Fehler werden durch eine Systemfehlermeldung angezeigt, die sich optisch vom automatischen Feedback unterscheidet. Solche Fehler können die Lernenden in der Regel leicht selbst beheben, wenn die Lehrkraft sie auf die in der Aufgabenstellung neben dem Schraubenschlüsselsymbol (vgl. Tabelle 8.1) angegebenen korrekten Schreibweisen hinweist.

8.1.2 Tipps

Die Lernenden werden bei vielen Aufgaben durch Tipps, die im Lernmaterial durch das Symbol des Rettungsrings gekennzeichnet sind, bei Bedarf unterstützt. Kleine-re Tipps können direkt unter der Aufgabenstellung ausgeklappt werden, während ausführliche Tipps auf einem separaten Jupyter Notebook gegeben werden. Dieses öffnet sich per Anklicken des bei der jeweiligen Aufgabe hinterlegten Links neben dem eigentlichen Arbeitsblatt. So können die Lernenden selbst entscheiden, ob und wann sie Unterstützung erhalten wollen. Die Hilfen bieten je nach Schwierigkeit der Aufgabe verschiedene Hinweise, Skizzen, Gleichungen oder erläuternde Beispiele. Manche Aufgaben bieten mehrere, gleichwertige Tipps, die unterschiedliche Aspekte

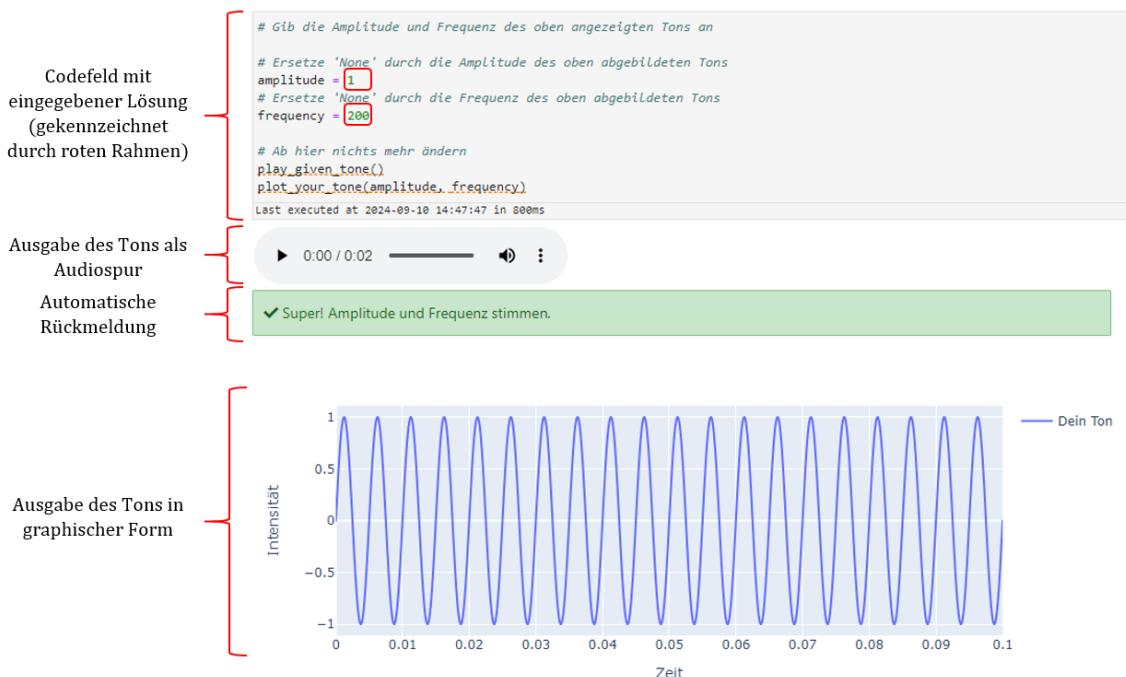


Abbildung 8.3: Bildschirmfoto einer Ausgabe auf einem digitalen Arbeitsblatt aus dem Lernmaterial zu Spracherkennung

der Aufgabe behandeln, sodass die Schüler gezielt die benötigte Hilfe auswählen können. Andere Aufgaben enthalten gestufte Tipps, die aufeinander aufbauend immer konkretere Hinweise geben.

Durch die digitalen Hilfen wird das Lernmaterial an die Bedürfnisse heterogener Lerngruppen angepasst, wodurch das Prinzip der Binnendifferenzierung realisiert wird (vgl. Kapitel 7, DP6). Zudem fördern die Tipps ein eigenständiges Arbeiten und ermöglichen eine Entlastung der Lehrkraft.

8.1.3 Vertiefendes Zusatz- und Informationsmaterial

Für beide Workshopthemen wurden ergänzende (optionale) Materialien erstellt, die bei Bedarf über Links auf den jeweiligen Arbeitsblättern aufgerufen werden können und sich in separaten Notebooks öffnen oder direkt über das Menü über den Workshopordner ausgewählt werden können. Die Zusatz- und Informationsmaterialien wurden entwickelt, um eine zusätzliche Differenzierung zu ermöglichen (vgl. Kapitel 7, DP6). Sie haben zwei unterschiedliche Ziele: Einige Informationsblätter sind darauf ausgelegt, mathematische Fähigkeiten oder Programmierfertigkeiten aufzufrischen oder zu erlernen, die für die Bearbeitung der Workshopmaterialien erforderlich sind (vgl. Abbildung 8.4). So werden Lernende, die beispielsweise noch keine Erfahrung im Programmieren haben, durch Informationen zu for-Schleifen oder if-Anweisungen unterstützt. Wer die Logarithmusgesetze noch nicht kennt, erhält eine kurze Einführung anhand eines Informationsblattes. Andere Informationsblätter

Logarithmusgesetze

Für das Rechnen mit Logarithmen gelten folgende Regeln:

Produktregel

Der Logarithmus eines Produktes ist die Summe der Logarithmen der beiden Faktoren:

$$\log(a \cdot b) = \log(a) + \log(b)$$

Quotientenregel

Der Logarithmus eines Bruchs ist der Logarithmus des Zählers abzüglich des Logarithmus des Nenners:

$$\log\left(\frac{a}{b}\right) = \log(a) - \log(b)$$

Potenzregel

Der Logarithmus einer Potenz ist das Produkt aus Exponenten und Logarithmus der Basis der Potenz:

$$\log(a^n) = n \cdot \log(a)$$

Abbildung 8.4: Bildschirmfoto eines Informationsblattes zu den Logarithmusgesetzen aus dem Lernmaterial zur Wortvorhersage

und die Zusatzmaterialien bieten weiterführende Einblicke, sowohl mathematischer als auch nicht mathematischer Art, die über den eigentlichen Workshop-Inhalt hinausgehen und nicht zwingend für die Bearbeitung der Lernmaterialien erforderlich sind. Diese zweite Funktion des vertiefenden Materials eignet sich insbesondere für sehr motivierte und interessierte Lernende. Zusätzlich zu den ausführlichen Zusatzblättern enthalten die Arbeitsblätter auch kleinere Zusatzaufgaben, die schnellere Lernende während oder nach der Bearbeitung der regulären Aufgaben bearbeiten können.

8.1.4 Antwortblätter

Für das Lernmaterial zur „Wortvorhersage“ werden insbesondere aufgrund der anzufertigenden visuellen Darstellungen mathematischer Objekte, zusätzlich zu den digitalen Arbeitsmaterialien, ausgedruckte Antwortblätter zur Verfügung gestellt. Auf diesen sollen die Lernenden Skizzen anfertigen und einzelne Textantworten festhalten. Die zugehörigen Aufgaben befinden sich auf den digitalen Arbeitsblättern. Das Stiftsymbol (vgl. Abbildung 8.1) weist auf die Bearbeitung auf einem beiliegenden Antwortblatt hin, sofern kein Texteingabefeld im digitalen Lernmaterial vorhanden ist. Die Antwortblätter sind im PDF-Format verfügbar und können sowohl auf einem digitalen Endgerät mit Zeichenfunktion als auch in ausgedruckter Form bearbeitet werden.

8.2 Aufbau des begleitenden Lehrmaterials

Das begleitende Lehrmaterial hat das Ziel, die eigenständige Durchführung durch Lehrkräfte im Unterricht bestmöglich zu unterstützen. Zum Begleitmaterial gehören

neben den Lösungen zu den digitalen Arbeitsblättern auch die Präsentationsfolien für die Einstiegs- und Sicherungsphasen der Workshops sowie ein ausführlich kommentierter Stundenverlaufsplan.

Auf der in Abschnitt 8.1 beschriebenen Workshopplattform liegt neben dem Lernmaterial auch das zugehörige Begleitmaterial. Die Lösungen werden anschließend in den Ordner „worksheets“ und das weitere Begleitmaterial in den Ordner „material_for_teachers“ geladen.

8.2.1 Präsentationsfolien

Für die Plenumsphasen stehen Präsentationsfolien zur Verfügung. Dazu zählen

- Folien für die Einstiegspräsentation, um eine Einführung in die Problemstellung zu geben,
- Folien für die Zwischenbesprechungen, um die erarbeiteten Teilergebnisse zu sichern, zu diskutieren und für den nächsten Modellierungsschritt Ideen im Plenum zu sammeln,
- Folien für die Abschlusspräsentation, um die finalen Ergebnisse zu besprechen, einen gesellschaftskritischen Blick auf das entwickelte System zu werfen, sowie ein Resumé zu ziehen.

In den Präsentationsfolien dient der Modellierungskreislauf als zentrales Orientierungselement, das genutzt wird, um in den Plenumsphasen die auf den Arbeitsblättern erarbeiteten Teilschritte in den Kreislauf einzuordnen.

8.2.2 Stundenverlaufsplan und Lösungen

Die Lösungen zu den Arbeitsblättern werden als separate Versionen der Arbeitsblätter mit eingetragenen Lösungen bereitgestellt. Diese sind daran zu erkennen, dass sie im Gegensatz zu den Schülerversionen keine „-SuS“-Endung im Dateinamen tragen. Zusätzlich wird ein ausführlich kommentierter Stundenverlaufsplan angefertigt. Der Stundenverlaufsplan soll der Lehrkraft helfen, die Unterrichtseinheit zu planen und durchzuführen. Er ordnet die einzelnen Phasen in einen zeitlichen Ablauf ein und gibt einen Überblick darüber, welche mathematischen Inhalte im Modellierungsprojekt thematisiert werden, welche Anbindung an den Schulunterricht besteht und welche Medien und Materialien in welcher Arbeitsphase benötigt werden. In dieser Arbeit ist zum Lernmaterial Spracherkennung (vgl. Tabelle 19.1) und zum Lernmaterial Wortvorhersage (vgl. Tabelle 13.1) ein Stundenverlaufsplan angefertigt worden. Zusätzlich werden das Vorwissen und die Einsatzmöglichkeiten des Lernmaterials benannt und erläutert, welche prozessbezogenen Kompetenzen durch das Lernmaterial besonders gefördert werden sollen.

9 Formate der Modellierungsprojekte und ihre üblichen Abläufe

9.1 Ablauf eines Modellierungstages

In diesem Kapitel wird der exemplarische Ablauf eines Modellierungstages beschrieben. Hierbei wird besonders auf diejenigen Elemente eingegangen, welche unabhängig vom inhaltlichen Thema des Modellierungstages strukturgebend für den Verlauf sind. Dazu zählen der Einstieg in die Problemstellung, eine Technik-Einführung, die Arbeitsphasen und Zwischensicherungen sowie die Abschlussdiskussion. Hierbei wird auf die zeitliche und methodische Abfolge des Unterrichtsganges sowie die verwendete Sozialform in den einzelnen Phasen eingegangen. Die Inhalte und konkreten Lernmaterialien der Modellierungstage zu den Themen „Wortvorhersage“ und „Spracherkennung“ werden in den nachfolgenden Kapiteln 13 und 19 erläutert.

Ein Modellierungstag umfasst in der Regel etwa fünf Zeitstunden. Diese Angabe, wie auch alle nachfolgenden, sind als Durchschnittswerte zu verstehen, die je nach Vorwissen, Motivation, Leistungsfähigkeit und Lernbereitschaft der Lerngruppe variieren können. Daher dienen diese Zahlen lediglich als Orientierung und müssen entsprechend den jeweiligen Bedürfnissen und Voraussetzungen angepasst werden.

Einführung in die Modellierung (15 Minuten, optional):

Der Modellierungstag beginnt mit einer Einführung in die Mathematische Modellierung, die über einen von dem Betreuenden gehaltenen Vortrag erfolgt. Der Prozess der mathematischen Modellierung wird am Beispiel eines vom Thema des Modellierungstages unabhängigen realen Problems eingeführt. Hierbei bietet sich ein leicht zugängliches motivierendes Problem an, an welchem sich die grundlegenden Überlegungen, die für die einzelnen Schritte des Modellierungsprozesses vorgenommen werden müssen, anschaulich exemplifizieren lassen. Ein Beispiel könnte die Thematisierung der Fragestellung „Wie muss ein Flugzeug gebaut werden, damit es möglichst weit fliegen kann?“ sein. Der in Abschnitt 2.1 in Abbildung 2.2 gezeigte vierstufige Modellierungskreislauf dient dabei als metakognitives Modell, das den Modellierungsprozess strukturiert und visualisiert. Die einzelnen Schritte des Modellierungsprozesses werden erläutert und in den Modellierungskreislauf eingeordnet. Zudem werden Einsatzgebiete der mathematischen Modellierung aufgezeigt. Diese Phase des Modellierungstages ist ein optionaler Bestandteil. Steht weniger Zeit zur Verfügung, so kann die Einführung der mathematischen Modellierung auch

direkt in der Einführung in die Problemstellung am Beispiel der Thematik des Modellierungstages erfolgen.

Einstieg in die Problemstellung (15 - 20 Minuten):

Nach dem Modellierungsvortrag führt die betreuende Person in einer weiteren Plenumsphase in die Problemstellung ein. Die zugehörigen Präsentationsfolien stehen Lehrpersonen auf der Lernplattform zur Verfügung (vgl. Abschnitt 8.2). Der Vortrag enthält in der Regel die wesentlichen Informationen, die die Lernenden als Grundlage für die Bearbeitung der Problemstellung benötigen. Um den Lernprozess zu fördern, wird der Vortrag möglichst interaktiv gestaltet, sodass individuelle Erfahrungen und eigene Ideen gebracht werden können. Eine Kurzfassung der zentralen Informationen aus dem Vortrag wird zu Beginn des ersten Arbeitsblatts bereitgestellt.

Einführung in den Umgang mit dem digitalen Lernmaterial (10 Minuten):

Nach der Einführung sollen die Lernenden selbstgesteuert mit dem digitalen Lernmaterial arbeiten. Dazu besuchen sie die Webseite der Workshopplattform und legen sich einen Account an. Anschließend kann das zum Workshop gehörige Lernmaterial über ein Drop-down-Menü ausgewählt werden. Dieser Vorgang wird von der betreuenden Person demonstriert, indem der Bildschirm am Beamer übertragen wird. Die Schülerinnen und Schüler führen parallel die Schritte aus und können bei Schwierigkeiten Fragen stellen. Nach dem Anmeldeprozess wird erläutert, wie die Arbeitsblätter strukturiert sind und wie ihre Bearbeitung erfolgt (vgl. Abschnitt 8.1).

Arbeitsphase mit Zwischensicherungen (ca. 4 Zeitstunden):

Die Bearbeitung des Lernmaterials erfolgt in Partnerarbeit (vgl. Kapitel 7, DP2). Während der Bearbeitung der Lernmaterialien erfolgt die Unterstützung der Lernenden nach dem Prinzip der minimalen Hilfe (Aebli, 2006, S. 300). Dies bedeutet, dass so viel Unterstützung wie nötig und gleichzeitig so wenig wie möglich geboten wird. Auf diese Weise wird man ermutigt, die Aufgaben eigenständig zu bearbeiten, eigene Lösungen zu finden und Fehler durch Diskussionen in Partnerarbeit selbst zu erkennen, ohne sich allein gelassen zu fühlen.

Nach jedem Arbeitsblatt erfolgt eine Zwischensicherung im Plenum. Der Zeitpunkt der Zwischensicherung sollte so gewählt werden, dass die meisten Lernenden das entsprechende Arbeitsblatt abgeschlossen haben. Besonders bei heterogenen Schülergruppen kann es trotz der Differenzierungsmöglichkeiten im Material zu unterschiedlichen Bearbeitungszeiten kommen. Dennoch sollte sichergestellt werden, dass zumindest der Großteil der Gruppe das Arbeitsblatt erfolgreich bearbeitet hat. In den Besprechungsphasen sollten die Lernenden die Gelegenheit haben, ihre Ergebnisse selbst mündlich vorzustellen (vgl. Kapitel 7 DP2). Idealerweise haben sie die Möglichkeit, ihren Bildschirm am Beamer zu teilen, um ihre Erläuterungen visuell zu unterstützen. Zugleich wird der auf dem Arbeitsblatt erarbeitete Inhalt in den

Gesamtlauf eingeordnet. Während des gesamten Modellierungsworkshops dient hierfür der Modellierungskreislauf als zentrales Orientierungselement. Die auf den Arbeitsblättern erarbeiteten Teilschritte werden in jeder Plenumsphase in den Kreislauf eingeordnet. Dies hilft den Lernenden, die einzelnen Inhalte in den Gesamtkontext des Modellierungsprozesses zu integrieren. Wie von Schukajlow et al. (2011, S. 42 - 43) vorgeschlagen, fungiert der Modellierungskreislauf als metakognitives, visuelles Hilfsmittel, das die Reflexion des Modellierungsprozesses durch die Lernenden unterstützt und so den Aufbau von Metawissen zum Modellierungskreislauf fördert.

Im Anschluss an eine Zwischensicherung erfolgt in manchen Fällen eine kurze Brainstormingphase, in welcher Ideen für den nächsten Modellierungsschritt gesammelt werden (vgl. Kapitel 7 DP2). Diese Phase zählt zwar nicht als Sicherung, sie wird in den Stundenverlaufsplänen jedoch wegen ihres kurzen zeitlichen Umfangs nicht separat aufgeführt, sondern zur Sicherungsphase hinzugerechnet.

Eine Zwischensicherung nach jedem Arbeitsblatt durchzuführen, bietet mehrere Vorteile. So kann zum einen abgeprüft werden, ob die festgelegten Teilziele des Arbeitsblatts erreicht wurden oder ob noch Unklarheiten bestehen, die geklärt werden müssen. Zum anderen können die gelernten Inhalte durch eine kurze Zusammenfassung gefestigt und gegebenenfalls ergänzt werden. Zudem wird der in dem Arbeitsblatt bearbeitete Teilschritt in den Modellierungskreislauf eingeordnet, wodurch der Gesamtzusammenhang verdeutlicht wird. Die regelmäßige Zwischensicherung nach jedem Arbeitsblatt stellt zudem einen methodischen Wechsel dar und sorgt so für Abwechslung im Lernprozess.

Abschlussdiskussion und Tagesabschluss (15 - 20 Minuten):

Am Ende des Modellierungstages reflektieren die Teilnehmer gemeinsam den gesamten Modellierungsprozess in einer Abschlussdiskussion. Die erzielten Ergebnisse werden dabei im Kontext des realen Problems interpretiert und validiert, sowie die Grenzen des Modells diskutiert. Im Anschluss daran erfolgt eine Auseinandersetzung mit gesellschaftlichen Fragestellungen zu den Nutzen und Risiken von Assistenzsystemen, die im offenen Austausch mit den Lernenden besprochen werden.

Zur weiteren Verbesserung der Workshops wird den Schülerinnen und Schülern ein schriftlicher Online-Fragebogen zur Verfügung gestellt, in dem sie ihre Erfahrungen und Meinungen zu Inhalt, Aufbau und Betreuung des Modellierungstags äußern können.

Der hier vorgestellte Ablauf eines Modellierungstages eignet sich insbesondere für außerschulische Lernprojekte oder im schulischen Kontext als Projekttag. Statt das Material in einem zusammenhängenden Format während eines Modellierungstages zu behandeln, ist es jedoch auch möglich, dieses als Unterrichtsreihe in den Schulunterricht zu integrieren. Die oben dargestellten Phasen bleiben erhalten, das Lernmaterial wird dafür lediglich in einzelne Unterrichtsstunden unterteilt. Die Zwischenpräsentationen können als Wiederholung zu jedem Stundenanfang genutzt werden. Zudem sollten zusätzliche Sicherungsphasen am Stundenende eingeplant werden,

falls es zeitlich nicht passt, dass die Stunde mit einer Zwischensicherung beendet wird.

Der Modellierungstag kann je nach Thema bei mehr Zeit um weitere Arbeitsphasen ergänzt werden. Es können zum einen im Standardmaterial als Blackbox eingeführte Verfahren näher betrachtet und zum anderen weiterführende Teilprobleme bearbeitet werden. Manche der Teilprobleme werden den Lernenden ohne kleinschrittige Aufgabenstruktur als offene Aufgabe gegeben. Für die Bearbeitung der offenen Aufgaben wird die Bearbeitung in Kleingruppen und eine intensive Betreuung empfohlen. Nähere Erläuterungen zum weiterführenden Material finden sich in Abschnitt 13.2.7.

9.2 Ablauf einer Modellierungs-Doppelstunde

In diesem Abschnitt soll der exemplarische Ablauf einer Doppelstunde zu einem Modellierungsproblem dargelegt werden. Hierbei werden besonders diejenigen Elemente hervorgehoben, welche sich von denen eines Modellierungstages unterscheiden. Die Inhalte und Lernmaterialien der Modellierungs-Doppelstunde zum Thema „Wortvorhersage“ werden in Abschnitt 13.2.8 erläutert. Die Dauer einer Modellierungs-Doppelstunde beträgt insgesamt 1,5 Zeitstunden und ist damit so lang wie eine typische Doppelstunde im Regelunterricht. Die im folgenden angegebenen Zeiten für die einzelnen Phasen der Doppelstunde sind auch hier Durchschnittswerte, die auf Erfahrungswerten mit Lernenden beruhen und je nach Lerngruppe abweichen können.

Einstieg in die Problemstellung (15 Minuten):

Diese Phase unterscheidet sich nicht von der eines Modellierungstages. Für eine detaillierte Beschreibung sei daher auf Abschnitt 9.1 verwiesen.

Arbeitsphase mit Zwischensicherungen (ca. 1 Zeitstunde):

Auch diese Phase ist sehr ähnlich zu derjenigen eines Modellierungstags. Im Vergleich zum Modellierungstag wird in der Doppelstunde jedoch mit analogen Arbeitsblättern gearbeitet, und es werden hier dementsprechend zunächst nur kleine Datensätze betrachtet. Um das entwickelte Modell jedoch sinnvoll erproben zu können, ist auch die Verarbeitung größerer Datenmengen und daher Computereinsatz notwendig. Das mit einem größeren Datensatz trainierte Modell wird deshalb zur Erprobung in einer Webanwendung bereitgestellt. Diese Webanwendung ist, wie auch die Lernplattform, über einen Webbrower über <https://e29208af.ka.bw-cloud-instance.org> zugänglich.

Abschlussdiskussion und Tagesabschluss (15 Minuten):

Auch diese Phase findet genauso statt wie am Modellierungstag. Hier wird daher ebenfalls für eine nähere Erläuterung auf Abschnitt 9.1 verwiesen.

Auch die Modellierungs-Doppelstunde kann bei mehr Zeit um weitere Arbeitsphasen zu einem weiterführenden Teilproblem ergänzt werden. Dieses wird ebenfalls auf zusätzlichen, analogen Arbeitsblättern bearbeitet.

9.3 Vergleich der Formate

Sowohl die Modellierungs-Doppelstunde als auch der Modellierungstag bieten wertvolle Ansätze zur Modellierung von realen Problemen mit Lernenden, unterscheiden sich jedoch deutlich in Bezug auf ihre Einsetzbarkeit, inhaltliche Tiefe und methodische Gestaltung. Im Folgenden werden zentrale Aspekte dieser beiden Workshopformate gegenübergestellt.

In der Doppelstunde bleibt der Fokus notwendigerweise auf der Entwicklung eines ersten einfachen Modells. Modellverbesserungen werden nur angedacht, die Mathematik hinter den Modellverbesserungen kann nicht betrachtet werden. Aufgrund des engen zeitlichen Rahmens ist es außerdem nicht möglich, die Modelle systematisch zu testen. Sie werden lediglich stichprobenartig erprobt. Der enge zeitliche Rahmen hat auch zur Folge, dass nur einfach zugängliche Problemstellungen bearbeitet werden können, die entweder stark vereinfacht oder insgesamt bereits leicht lösbar sind. Im Modellierungstag hingegen kann die Problemstellung umfassender und inhaltlich tiefer betrachtet werden. Hier ist ausreichend Zeit, um neben der Modellentwicklung auch die Evaluation und Verbesserung der Modelle sowie deren zugrunde liegenden mathematischen Konzepte zu beleuchten.

Ein weiterer Vorteil des Modellierungstages liegt in der höheren Motivation der Lernenden. Da dieses Format den üblichen Schulrhythmus durchbricht und eine intensivere Auseinandersetzung mit dem Thema ermöglicht, kann es als spannender und abwechslungsreicher empfunden werden. Im Gegensatz dazu ist die Modellierungs-Doppelstunde Teil des regulären Unterrichtsverlaufs, in dem die Fächer nach etwa 90 Minuten wechseln, was die Möglichkeit für eine tiefergehende Beschäftigung mit einem Thema einschränkt.

Die Anforderungen an die Schülerinnen und Schüler variieren ebenfalls zwischen den beiden Formaten. In der Doppelstunde müssen die Lernenden schnell in die Problemstellung und den zugehörigen Kontext einsteigen. Beim Modellierungstag hingegen ist mehr Durchhaltevermögen gefragt, da die intensive Beschäftigung mit einer komplexeren Problematik über einen längeren Zeitraum Konzentration und Ausdauer erfordert.

Während in der Doppelstunde eher geschlossene Aufgabenformate enthalten sind, bietet der Modellierungstag die Möglichkeit, neben den geschlossenen Aufgaben auch einzelne offene Aufgaben zu stellen. Die Schülerinnen und Schüler können dabei individuell und selbstgesteuert arbeiten, was eine natürliche Differenzierung fördert. Werden offene Aufgaben gestellt, erfordert dies allerdings auch einen höheren Betreuungsschlüssel, da die Lernenden bei der eigenständigen Bearbeitung mehr Unterstützung benötigen. Die Selbstständigkeit im Umgang mit neuen Aufgabenformaten kann jedoch ihre Fähigkeit stärken, mathematische Probleme kreativ zu lösen.

Ein weiterer Unterschied liegt im Einsatz digitaler Medien. In der Modellierungs-Doppelstunde wird größtenteils auf digitale Medien verzichtet, da die Einarbeitung in ein neues Tool wertvolle Zeit in der ohnehin begrenzten Unterrichtsdauer beanspruchen würde. Allerdings ist der völlige Verzicht auf digitale Hilfsmittel problematisch.

Für die Lösung realer Probleme ist die Verarbeitung größerer Datenmengen nämlich meist notwendig. Um dennoch das Erproben des auf Basis realer Daten entwickelten Modells zu ermöglichen, wird eine Webanwendung bereitgestellt. Das zugrunde liegende Modell basiert auf den gemeinsam erarbeiteten mathematischen Konzepten, die Implementierung der mathematischen Methoden geschieht allerdings im Hintergrund. Im Modellierungstag hingegen spielt der Einsatz digitaler Medien eine größere Rolle, da ausreichend Zeit vorhanden ist, um sich mit diesen vertraut zu machen und sie effektiv zu nutzen. Sind Lernende bereits im Umgang mit Jupyter-Notebooks vertraut, so ist deren Nutzung als digitale Lernumgebung auch für Doppelstunden denkbar.

Im Hinblick auf den Regelunterricht ist die Modellierungs-Doppelstunde unter den derzeit vorherrschenden Rahmenbedingungen an den Schulen einfacher zu integrieren als der Modellierungstag. Grund dafür ist unter anderem der kürzere Zeitrahmen. Dies führt dazu, dass sich die Modellierungs-Doppelstunde neben den anderen Inhalten, die während eines Schuljahres thematisiert werden müssen, eher unterbringen lässt und durch das zeitliche Format auch besser in den regulären Stundenablauf passt. Eng geknüpft an die zeitliche Restriktion ist auch der stärker begrenzte thematische Umfang. Dieser führt außerdem dazu, dass sich die Lehrkraft für eine Modellierungs-Doppelstunde weniger tief in die Problemstellung einarbeiten muss, was die Organisation erleichtert und eine Einordnung in den curricularen Ablauf vereinfacht.

Ein Modellierungstag, der deutlich mehr Zeit in Anspruch nimmt, stellt hingegen aufgrund der schulischen Zeitrestriktionen und des umfassenderen Themenspektrums eine größere Herausforderung dar. Hier wird der Unterrichtsalltag stärker durchbrochen, was in vielen Fällen nur schwer mit dem regulären Stundenplan vereinbar ist.

Zusammenfassend lässt sich sagen, dass die Modellierungs-Doppelstunde eine praktikable Lösung für den Regelunterricht darstellt, während der Modellierungstag tiefere Einblicke in mathematische Modellierungsprozesse bietet und die Motivation stärker fördern kann. Ein wünschenswerter Schritt wäre es, die Rahmenbedingungen im Schulunterricht so anzupassen, dass projektartige Formate wie der Modellierungstag zukünftig auch im Regelunterricht vermehrt Raum finden können.

Teil III

Lehr- und Lernmaterial zur Wortvorhersage

10 Einleitung

Ein alltagsnahes Beispiel, um mit Schülerinnen und Schülern in das Thema NLP einzusteigen, bieten Assistenzsysteme, welche Smartphone-nutzende beim Tippen unterstützen. Diese Assistenzsysteme werden im Alltag vieler Lernender regelmäßig verwendet, wenn sie mit Freunden chatten. Zu diesen Assistenzsystemen zählen zum Beispiel die Vervollständigung von Wörtern, welche der/ die Nutzende bereits teilweise eingetippt hat, sowie automatische Korrekturvorschläge für möglicherweise fehlerhaft eingetippte Wörter (vgl. Abbildung 10.1). Auch die Wortvorhersage zählt zu den typischen Assistenzsystemen, welche am Smartphone zum Einsatz kommen. Sie unterstützt Nutzende beim Schreiben einer Nachricht durch eine Auswahl an Vorschlägen, häufig drei verschiedene, für das nächste Wort (vgl. Abbildung 10.1). Durch die Nutzung dieser Assistenzsysteme werden nicht nur Zeit gespart und weniger fehlerhafte Texte getippt, sondern es wird auch die kognitive Hürde beim Schreiben gesenkt. Die genannten Assistenzsysteme finden neben der Nutzung am Smartphone aber auch in anderen Anwendungsbereichen Einsatz. Dazu gehören zum Beispiel Eingabefelder wie die URL-Leiste oder Adressfelder, aber auch die Suchanfrage im Browser (Nandi & Jagadish, 2007, S. 219).

In dieser Arbeit wird die Funktionsweise der Wortvorhersage und deren Einsatz als Unterstützung beim Tippen am Smartphone untersucht. Um Wortvorschläge zu ermitteln, muss zunächst ein Modell der (schriftlichen) Sprache erstellt werden, welches Abhängigkeiten einzelner Satzbausteine untereinander darstellen kann. Dieses Sprachmodell kann dann genutzt werden, um diejenigen Wörter, die besonders gut zum vorangegangenen Satzkontext passen, ausfindig zu machen und vorzuschlagen.

Dieser Teil der Arbeit führt eine stoffdidaktische Analyse des Themas Wortvorhersage und den damit verbundenen mathematischen Inhalten durch. Die Analyse

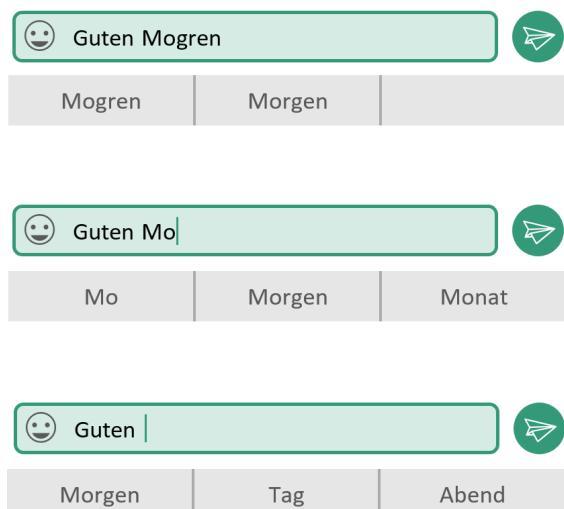


Abbildung 10.1: Anwendungsbeispiele unterschiedlicher Assistenzsysteme: Autokorrektur (oben), Wortvervollständigung (mittig), Wortvorhersage (unten)

mündet in der Entwicklung einer digitalen Lernumgebung, die mit Lernenden erprobt wird. Dazu wird in Kapitel 11 zunächst theoretisch erarbeitet, wie ein solches Sprachmodell entwickelt werden kann und welche zentralen mathematischen Konzepte eine Rolle spielen. In Kapitel 12 werden die erarbeiteten Methoden auf reale Trainingstexte angewendet und anschließend in Kapitel 13 didaktisch aufbereitet und als Lehr- und Lernmaterial eines Modellierungstags sowie einer Modellierungs-Doppelstunde umgesetzt. Kapitel 14 zeigt abschließend auf, welche Erfahrungen mit dem Einsatz des Lernmaterials gemacht wurden und zu welchen Anpassungen dies am Material führte. Zudem wird in Kapitel 14 erläutert, wie die zuvor in Kapitel 7 formulierten Design-Prinzipien im Lernmaterial umgesetzt werden und damit unter anderem dargelegt, weshalb sich die Wortvorhersage als Problemstellung für den mathematischen Modellierungsunterricht zum Kontext NLP eignet. Die folgenden Kapitel greifen damit Inhalte auf, die von der Autorin bereits in den Veröffentlichungen Hofmann und Frank (2022), Hofmann und Frank (2023) sowie Hofmann und Frank (2024) thematisiert wurden, ergänzen und bündeln sie.

11 Theoretischer Hintergrund

Sprachmodelle dienen der Erfassung und Vorhersage von sprachlichen Strukturen. Die Anordnung von Wörtern in einem Satz erfolgt nicht willkürlich, sondern unterliegt bestimmten Abhängigkeiten, die entweder über syntaktische oder semantische Regeln gegeben sind (Euler, 2006, S. 112). In statistischen Sprachmodellen werden diese Abhängigkeiten durch Wahrscheinlichkeiten beschrieben. Kern eines statistischen Sprachmodells ist somit eine Wahrscheinlichkeitsverteilung, die den Wortsequenzen unterschiedliche Wahrscheinlichkeiten zuweist (Fink, 2003, S. 99). Diese Wahrscheinlichkeitsverteilungen bilden die Grundlage dafür, die Dynamik und Muster der Sprache zu erfassen und somit prädiktive Analysen, beispielsweise zur Vorhersage des nächsten Wortes beim Tippen, zu ermöglichen.

Ein sehr verbreitetes und gut erforschtes Beispiel eines statistischen Sprachmodells ist das *N-Gramm-Modell* (z.B. Euler, 2006, S. 112; Jurafsky & Martin, 2023; Ney et al., 1997, S. 175; Shannon, 1948, S. 387; Wendemuth, 2004, S. 25). Im Folgenden soll gezeigt werden, dass dessen grundlegende Funktionsweise auf basalnen mathematischen Methoden fundiert und sich das *N-Gramm-Modell* deshalb besonders für die Thematisierung im Mathematikunterricht eignet. Dabei stellt sich als besonders herausfordernd heraus, dass Veröffentlichungen zu *N-Gramm-Modellen* aus Sicht der Anwendungsforschung und nicht aus einer mathematischen Perspektive geschrieben sind. Der folgende Abschnitt legt zunächst die mathematischen Grundlagen dar und führt erste Überlegungen zur Zugänglichkeit für Lernende aus.

11.1 *N-Gramm-Modelle*

Um das Erzeugen von Sprache oder konkret von einer Wortsequenz mathematisch modellieren zu können, wird das Auswählen eines Wortes an jeder Position dieser Wortsequenz als Zufallsexperiment aufgefasst. Die Zufallsvariable X_i modelliere das ausgewählte Wort an der i -ten Position dieser Wortsequenz. Der Zustandsraum V mit $V = \{v_1, \dots, v_s\}$, auf den X_i abbildet, enthält s voneinander verschiedene Wörter. In der Sprachverarbeitung bezeichnet man diesen Zustandsraum auch als *Vokabular*. Der Grundraum enthält hierbei alle Wörter einer Sprache (einer bestimmten Textart). Dabei wird die idealisierende Annahme zugrunde gelegt, dass Sprache als statische Größe betrachtet werden kann. Obwohl Sprache einem stetigen Wandel unterliegt, hat sich diese Vereinfachung in der Praxis als nützlich erwiesen und wird daher bei der Konstruktion statistischer Sprachmodelle bewusst angewendet. Das Erzeugen einer Wortsequenz der Länge n kann dann als Realisierung eines stochastischen Prozesses (X_1, \dots, X_m) in diskreter Zeit modelliert werden. Für die

Modellierung wird das Konzept einer *Markov-Kette* verwendet (z.B. Euler, 2006, S. 113; Fink, 2003, S. 99; Nadas, 1984, S. 859; Ney et al., 1997, S. 175).

Eine (homogene) *Markov-Kette N-ter Ordnung* ist ein stochastischer Prozess, für den die *Markov-Eigenschaft*

$$\mathbb{P}(X_i = b_i | X_1 = b_1, \dots, X_{i-1} = b_{i-1}) = \mathbb{P}(X_i = b_i | X_{i-N} = b_{i-N}, \dots, X_{i-1} = b_{i-1}) \quad (11.1)$$

gilt. Dabei gelte $i \geq N + 1$, und b_1, b_2, \dots, b_i sind beliebige Zustände aus V mit $\mathbb{P}(X_{i-N} = b_{i-N}, \dots, X_{i-1} = b_{i-1}) > 0$ (Serfozo, 2009, S. 2). Hierbei sind X_1, \dots, X_n Zufallsvariablen auf einem gemeinsamen Wahrscheinlichkeitsraum (Ω, \mathbb{P}) mit $X_i : \Omega \rightarrow V$. Die Wahrscheinlichkeit $\mathbb{P}(X_{i-N} = b_{i-N}, \dots, X_{i-1} = b_{i-1})$ wird als *Übergangswahrscheinlichkeit* bezeichnet.

Bei einer Markov-Kette hängt die Verteilung des Zustands zur Zeit i also nur von den Zuständen zu den Zeiten $i-1$ bis $i-N$, also den N vorangegangenen Zuständen, ab. Die Markov-Kette heißt *homogen*, wenn sie *zeithomogen* ist, die Übergangswahrscheinlichkeiten somit nicht vom Zeitpunkt des Übergangs abhängen.

Das Erzeugen einer Wortsequenz soll nun mithilfe einer homogenen Markov-Kette modelliert werden. Dazu wird die Wahrscheinlichkeit

$$\mathbb{P}(X_1 = w_1, \dots, X_m = w_m)$$

für das Auftreten einer konkreten Wortsequenz w_1, \dots, w_m der Länge m mit $w_1, \dots, w_m \in V$ betrachtet. Mit der allgemeinen Multiplikationsregel (vgl. Henze, 2023, S. 135 - 136) gilt

$$\begin{aligned} \mathbb{P}(X_1 = w_1, \dots, X_m = w_m) &= \mathbb{P}(X_1 = w_1) \cdot \mathbb{P}(X_2 = w_2 | X_1 = w_1) \\ &\quad \cdot \dots \cdot \mathbb{P}(X_m = w_m | X_1 = w_1, \dots, X_{m-1} = w_{m-1}). \end{aligned} \quad (11.2)$$

Die Übergangswahrscheinlichkeiten in diesem Ausdruck geben an, mit welcher Wahrscheinlichkeit ein Wort auf eine auch *Wortvorgeschichte* genannte gegebene Wortsequenz folgt.

Bei dieser Faktorisierung können allerdings mit zunehmender Länge der Wortsequenz bedingte Wahrscheinlichkeiten mit beliebig langen Bedingungen entstehen. Um das Sprachmodell in der Praxis handhabbar zu machen, wird daher die Annahme getroffen, dass nur die letzten $N-1$ Wörter ausschlaggebend für das Auftreten des nächsten Wortes sind. (X_1, \dots, X_m) entspricht somit einer Markov-Kette $(N-1)$ -ter Ordnung mit der Markov-Eigenschaft (vgl. Gleichung (11.1))

$$\begin{aligned} &\mathbb{P}(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) \\ &= \mathbb{P}(X_i = w_i | X_{i-(N-1)} = w_{i-(N-1)}, \dots, X_{i-1} = w_{i-1}), \end{aligned}$$

wobei $i-(N-1) \geq 1$ gilt. In der Sprachverarbeitung ist für diese Markov-Kette auch die Bezeichnung *N-Gramm-Modell* üblich. Nimmt N den Wert zwei an, wird also das Auftreten eines Wortes einer Wortsequenz nur vom vorherigen Wort beeinflusst, so spricht man von einem *Bi-Gramm-Modell*. Für die Übergangswahrscheinlichkeit

gilt dann

$$\mathbb{P}(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) = \mathbb{P}(X_i = w_i | X_{i-1} = w_{i-1}) \quad (11.3)$$

mit $i \geq 2$. Der Ausdruck in (11.2) vereinfacht sich damit zu

$$\begin{aligned} \mathbb{P}(X_1 = w_1, \dots, X_m = w_m) = & \mathbb{P}(X_1 = w_1) \cdot \mathbb{P}(X_2 = w_2 | X_1 = w_1) \\ & \cdot \dots \cdot \mathbb{P}(X_m = w_m | X_{m-1} = w_{m-1}). \end{aligned} \quad (11.4)$$

Wird die Annahme getroffen, dass das Auftreten eines Wortes nur von den beiden vorangegangenen Wörtern beeinflusst wird, so bezeichnet man das zugehörige Modell als *Tri-Gramm-Modell*. Für dieses gilt also

$$\mathbb{P}(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) = \mathbb{P}(X_i = w_i | X_{i-2} = w_{i-2}, X_{i-1} = w_{i-1})$$

mit $i \geq 3$. Beim *Uni-Gramm-Modell* wird das Auftreten eines Wortes als gänzlich unabhängig von der Wortvorgeschichte angenommen, und es gilt

$$\mathbb{P}(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) = \mathbb{P}(X_i = w_i) \quad (11.5)$$

mit $i \geq 1$. Je größer N ist, desto besser lässt sich die Abhängigkeit von den vorangegangenen Wörtern und damit der Satzkontext modellieren.

Das N -Gramm-Modell weist bei einer gegebenen Wortsequenz jeder Möglichkeit für das nachfolgende Wort eine Wahrscheinlichkeit zu. Ein Bi-Gramm-Modell legt beispielsweise für jedes Wort im Vokabular die Wahrscheinlichkeit fest, mit der ein anderes Wort darauf folgt. Ordnet man die einzelnen Wörter als Knoten in einem gerichteten Graphen an, so können die Übergangswahrscheinlichkeiten als die Wahrscheinlichkeiten für Übergänge zwischen Knoten angesehen werden. Bei einem Tri-Gramm-Modell werden entsprechend zwei Wörter zu einem Knoten zusammengefasst und auch hier wird die Wahrscheinlichkeit für einen Übergang von einem zum anderen Knoten über die Übergangswahrscheinlichkeit des Tri-Gramm-Modells angegeben. Diese Graphen werden auch als *Übergangsgraphen* bezeichnet. Beispiele für Übergangsgraphen finden sich unter anderem in der Erläuterung des Lernmaterials in Abbildung 13.2.

Eine Anwendung, die N -Gramm-Modelle nutzt, ist beispielsweise die Vorhersage für das nächste Wort beim Chatten am Smartphone oder beim Eintippen von Suchanfragen. Die Vorhersage des nächsten Wortes erfolgt nach der Größe der Übergangswahrscheinlichkeit. Eine Auswahl an wenigen – häufig drei – Wörtern mit den größten Übergangswahrscheinlichkeiten werden als Vorschlag präsentiert.

Die Anwendung von Markov-Ketten zur Beschreibung von statistischen Zusammenhängen in Texten wurde bereits vom russischen Mathematiker Andrej Andrejewitsch Markov und Namensgeber der Markov-Kette 1913 vorgeschlagen (Von Hilgers & Langville, 2006, S. 156). Er verwendete erstmals eine solche Modellierung zur statistischen Analyse der Buchstabenabfolge im Text von Alexander Puschkins Gedicht „Eugen Onegin“ (Markov, 1913). Claude Shannon griff diese Idee auf und erweiterte sie, indem er Markov-Ketten nicht nur zur Analyse stochastischer Ereignisse, sondern

auch zur Generierung solcher Ereignisse verwendete. In seiner Arbeit zeigte Shannon (1948, S. 384 - 389), dass Buchstabenfolgen, die durch ein Tri-Gramm-Modell erzeugt werden, dessen Tri-Gramme aus drei aufeinanderfolgenden Buchstaben bestehen, Merkmale natürlicher Sprache aufweisen können. Die Übergangswahrscheinlichkeiten schätzte er über die Auftretenshäufigkeit von Wortsequenzen in englischen Texten und erzeugte so folgende Sequenz:

„THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED“
 (Shannon, 1948, S. 388).

Kernaspekte: Um die Auswahl des nächsten Wortes als Zufallsexperiment modellieren zu können, muss Sprache vereinfacht als statisch angenommen werden. Das Erzeugen eines Textes kann dann als endlicher stochastischer Prozess modelliert werden. Um das Modell handhabbar zu machen, beschränkt man die Abhängigkeit des Auftretens eines Wortes auf die letzten $N - 1$ Ereignisse.

11.2 Schätzen der Übergangswahrscheinlichkeiten

Um die Übergangswahrscheinlichkeiten für die Wortvorhersage beim Chatten zu schätzen, greift man in der Sprachverarbeitung üblicherweise auf bereits vorhandene Texte zurück, deren Wortsequenzen auf ihre Häufigkeit hin untersucht werden (Ney et al., 1997, S. 179). Dabei wird angenommen, dass diese, auch als *Trainingsdatensatz* bezeichneten, sogenannten *Trainingstexte* eine zufällige Stichprobe des deutschen Sprachgebrauchs (aller Texte die in Alltagssprache geschrieben sind) darstellen und somit die Grundgesamtheit hinreichend gut repräsentieren. Diese Annahme ist insbesondere für stark flektierende Sprachen wie das Deutsche mit seinen vielen Wortformen und Neuschöpfungen durch Wortzusammensetzungen nicht umstritten. Beim N -Gramm-Modell wird sie jedoch in Kauf genommen, um Übergangswahrscheinlichkeiten schätzen und damit ein mathematisches Modell bilden zu können. Zusammenfassend sind für die Entwicklung eines N -Gramm-Modells somit drei elementare idealisierende Annahmen grundlegend:

- Sprache wird als statische Größe angesehen.
- Die Trainingstexte repräsentieren den Sprachgebrauch (aller Texte die in Alltagssprache geschrieben sind) ausreichend gut.
- Nur die $N - 1$ letzten Wörter beeinflussen das Auftreten des nächsten Wortes.

In den nachfolgenden Abschnitten soll gezeigt werden, dass die relative Auftretenshäufigkeit einer Wortsequenz in einem Trainingstext ein Schätzwert für die Übergangswahrscheinlichkeit ist. Konkret bedeutet das, dass für das Uni-, Bi- und Tri-Gramm-Modell mit den nicht von i abhängenden Kurzschreibweisen

$$p(v_k) := \mathbb{P}(X_i = v_k) \quad \forall i = 1, \dots, m \quad (11.6)$$

$$p(v_k|v_j) := \mathbb{P}(X_i = v_k | X_{i-1} = v_j) \quad \forall i = 2, \dots, m \quad (11.7)$$

$$p(v_k|v_\ell, v_j) := \mathbb{P}(X_i = v_k | X_{i-2} = v_\ell, X_{i-1} = v_j) \quad \forall i = 3, \dots, m \quad (11.8)$$

für alle $v_k, v_\ell, v_j \in V$ die jeweiligen Schätzwerte durch

$$\begin{aligned}\hat{p}(v_k) &= \frac{c(v_k)}{\sum_{r=1}^s c(v_r)} \quad \forall k = 1, \dots, s \\ \hat{p}(v_k|v_j) &= \frac{c(v_j, v_k)}{\sum_{r=1}^s c(v_j, v_r)} \quad \forall k, j = 1, \dots, s \\ \hat{p}(v_k|v_\ell, v_j) &= \frac{c(v_\ell, v_j, v_k)}{\sum_{r=1}^s c(v_\ell, v_j, v_r)} \quad \forall k, \ell, j = 1, \dots, s\end{aligned}$$

gegeben sind. Dabei bezeichnen $c(v_k), c(v_j, v_k), c(v_\ell, v_j, v_k)$ Auftretenshäufigkeiten der Wortsequenzen v_k , bzw. v_j, v_k , bzw. v_ℓ, v_j, v_k .

Im Lernmaterial werden die Schätzwerte zwar nicht hergeleitet, sondern es wird intuitiv mit den relativen Auftretenshäufigkeiten gearbeitet. Trotzdem soll hier gezeigt werden, dass es mathematisch begründbar ist, weshalb die Übergangswahrscheinlichkeit über die relative Auftretenshäufigkeit geschätzt werden kann. Darüber hinaus wird später Abschnitt 11.4 gezeigt, dass die bei der Herleitung genutzte *Maximum-Likelihood-Schätzmethode* auch eine zentrale Rolle bei der Bewertung von Sprachmodellen spielt. Die nachfolgenden beiden Abschnitte zeigen daher eine Herleitung der Schätzwerte für das Uni-Gramm-Modell und das Bi-Gramm-Modell. Dabei wird der Beweis zur Schätzung von Parametern der Multinomialverteilung verwendet, welcher bei Rao (1973, S. 355 - 357) und in didaktisch aufbereiteter Form bei Henze (2025) zu finden ist. Für N -Gramm-Modelle höherer Ordnung kann man auf die gleiche Weise vorgehen.

11.2.1 Maximum-Likelihood-Schätzer für das Uni-Gramm-Modell

Zunächst soll über die Maximum-Likelihood-Schätzmethode der Maximum-Likelihood-Schätzwert (kurz ML-Schätzwert) für die Übergangswahrscheinlichkeiten des Uni-Gramm-Modells $p(v_i)$ mit $v_i \in V$ ermittelt werden. Es sollen diejenigen Übergangswahrscheinlichkeiten gefunden werden, unter welchen der Trainingsdatensatz die größte Auftretenswahrscheinlichkeit besitzt. Dazu wird das Uni-Gramm-Modell als Multinomialexperiment mit den Ausgängen v_1, \dots, v_s interpretiert, welches m -mal in unabhängiger Folge durchgeführt wird. Die Zufallsvariable Y_k beschreibe dabei die vor der Durchführung des Experiments zufällige Anzahl der Ausgabe v_k ($k = 1, \dots, s$). In einem theoretischen Modell gibt es Wahrscheinlichkeiten $p_1 := p(v_1), \dots, p_s := p(v_s)$ mit $p_k \geq 0$ für jedes k und $\sum_{k=1}^s p_k = 1$ für die Ausgänge dieses Multinomialexperiments. Die Wahrscheinlichkeit, dass für jedes k der

Ausgang v_k genau $c_k := c(v_k)$ mal auftritt, ist dann durch

$$\mathbb{P}(Y_1 = c_1, \dots, Y_s = c_s) = \frac{m!}{c_1! \cdot \dots \cdot c_s!} p_1^{c_1} \cdot \dots \cdot p_s^{c_s}$$

gegeben. Seien tatsächlich Trainingsdaten mit Auftretenshäufigkeiten c_1, \dots, c_s mit $c_1 + \dots + c_s = m$ beobachtet worden. Es sind diejenigen Parameter p_1, \dots, p_s gesucht, für die diese Wahrscheinlichkeit maximal wird. Um darzustellen, dass die Verteilung des Zufallsvektors (Y_1, \dots, Y_s) abhängig von der Wahl der Parameter p_1, \dots, p_s ist, wird die Verteilung mit p_1, \dots, p_s indiziert und

$$\mathcal{L}_{c_1, \dots, c_s}(p_1, \dots, p_s) := \mathbb{P}_{c_1, \dots, c_s}(Y_1 = c_1, \dots, Y_s = c_s) = \frac{m!}{c_1! \cdot \dots \cdot c_s!} p_1^{c_1} \cdot \dots \cdot p_s^{c_s}$$

als Likelihood-Funktion zu den beobachteten Ausgängen c_1, \dots, c_s definiert. Da der Multinomialkoeffizient nicht von den Parametern p_k abhängt, wird

$$\mathcal{L}_{c_1, \dots, c_s}^*(p_1, \dots, p_s) := p_1^{c_1} \cdot \dots \cdot p_s^{c_s}$$

definiert, und es ist das Maximum dieser Funktion unter den Nebenbedingungen

$$p_1, \dots, p_s \geq 0, \quad \sum_{k=1}^s p_k = 1 \tag{11.9}$$

gesucht. Weil der Logarithmus eine streng monoton wachsende Funktion ist und jedes Maximum des Logarithmus der Likelihood-Funktion daher auch ein Maximum der Likelihood-Funktion darstellt, ist äquivalent dazu das Maximum von

$$\log \mathcal{L}_{c_1, \dots, c_s}^*(p_1, \dots, p_s) := \log (p_1^{c_1} \cdot \dots \cdot p_s^{c_s}) = \sum_{k=1}^s c_k \cdot \log p_k$$

unter den Nebenbedingungen (11.9) gesucht.¹ Die folgenden Überlegungen zeigen, dass in obiger Situation $(\widehat{p}_1, \dots, \widehat{p}_s) = \left(\frac{c_1}{m}, \dots, \frac{c_s}{m}\right)$ der Maximalwert ist. Es gilt

$$\begin{aligned} \log \mathcal{L}_{c_1, \dots, c_s}^*(p_1, \dots, p_s) &= \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \log p_k \\ &= \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \log \left(p_k \cdot \frac{m}{c_k} \cdot \frac{c_k}{m} \right) \\ &= \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \left(\log \frac{p_k m}{c_k} + \log \frac{c_k}{m} \right). \end{aligned}$$

¹Da vorausgesetzt wurde, dass gewisse der p_k gleich null sein können, kann es passieren, dass im Logarithmus der Likelihood-Funktion der Faktor $\log 0 = -\infty$ auftritt. Aufgrund des Satzes von l'Hospital gilt aber $\lim_{x \rightarrow 0} x \cdot \log x = 0$.

Aufgrund der Gültigkeit der Logarithmusungleichung $\log x \leq x - 1$ für $x > 0$ gilt $\log \frac{p_k m}{c_k} \leq \frac{p_k m}{c_k} - 1$, da $\frac{p_k m}{c_k} > 0$ falls $c_k > 0$ wegen (11.9). Damit folgt

$$\begin{aligned}
 \log \mathcal{L}_{c_1, \dots, c_s}^*(p_1, \dots, p_s) &\leq \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \left(\frac{p_k m}{c_k} - 1 + \log \frac{c_k}{m} \right) \\
 &= \sum_{\substack{k=1 \\ c_k > 0}}^s p_k m - \sum_{\substack{k=1 \\ c_k > 0}}^s c_k + \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \log \frac{c_k}{m} \\
 &= m \cdot \sum_{\substack{k=1 \\ c_k > 0}}^s p_k - m + \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \log \frac{c_k}{m} \\
 &\leq \sum_{\substack{k=1 \\ c_k > 0}}^s c_k \cdot \log \frac{c_k}{m} = \log \mathcal{L}_{c_1, \dots, c_s}^* \left(\frac{c_1}{m}, \dots, \frac{c_s}{m} \right).
 \end{aligned}$$

Dabei verwendete die letzte Abschätzung, dass die Summe über alle p_k abzüglich derer, für die $c_k = 0$ gilt, kleiner gleich eins ist. Damit ist

$$\hat{p}_k = \frac{c_k}{m} = \frac{c(v_k)}{\sum_{r=1}^s c(v_r)} \quad \forall k = 1, \dots, s \quad (11.10)$$

ML-Schätzwert für die Übergangswahrscheinlichkeit p_k beim Uni-Gramm-Modell zu den beobachteten Ausgängen c_1, \dots, c_s . Dieser ist also die beobachtete relative Auftretenshäufigkeit des Wortes v_k .

Um die Genauigkeit dieses Schätzwertes für die Übergangswahrscheinlichkeit zu untersuchen, wird c_k als Realisierung der Zufallsvariable Y_k betrachtet. Der Schätzwert \hat{p}_k ist aufgrund der Tatsache, dass er auf der Anzahl der Ausgänge c_k aus vorliegenden Daten in m Versuchen basiert, eine Realisierung der Zufallsvariable $R_k := \frac{Y_k}{m}$, die als zufälliger relativer Anteil der beobachteten Ausgänge k -ter Art in der Stichprobe definiert ist. Diese Zufallsvariable wird auch als *Schätzer* für p_k bezeichnet. Für den Erwartungswert und die Varianz von R_k gelten

$$\begin{aligned}
 \mathbb{E}_{p_k}(R_k) &= \mathbb{E}_{p_k} \left(\frac{Y_k}{m} \right) = \frac{1}{m} \mathbb{E}_{p_k}(Y_k) = \frac{1}{m} m p_k = p_k, \\
 \mathbb{V}_{p_k}(R_k) &= \mathbb{V}_{p_k} \left(\frac{Y_k}{m} \right) = \frac{1}{m^2} \mathbb{V}_{p_k}(Y_k) = \frac{1}{m^2} m p_k (1 - p_k) = \frac{p_k (1 - p_k)}{m}.
 \end{aligned}$$

Dabei sind sowohl die Eigenschaften von Erwartungswert und Varianz eingegangen, als auch die Tatsache, dass die Zufallsvariable Y_k binomialverteilt ist, da hier statt des gesamten Multinomialexperiments nur die Ausgänge k -ter Art beziehungsweise nicht k -ter Art relevant sind. Der Erwartungswert und die Varianz wurden dann unter der Annahme, dass p_k der wahre Parameter der Verteilung von Y_k ist und somit $Y_k \sim \text{Bin}(m, p_k)$ gilt, bestimmt. Unabhängig vom tatsächlich zugrunde liegenden Parameter p_k zeigt sich, dass der Erwartungswert des Schätzers genau p_k entspricht. Der Schätzer ist daher erwartungstreu.

Eine weitere Einsicht ergibt sich aus der Varianz des Schätzers. Diese nimmt unabhängig vom tatsächlichen Wert von p_k mit wachsender Stichprobengröße – also mit zunehmender Länge des vorliegenden Trainingsdatensatzes – ab, und bestätigt damit, was auch intuitiv sinnvoll erscheint: Je größer der Trainingsdatensatz, desto häufiger können einzelne Wortsequenzen beobachtet werden und desto genauer ist die relative Auftretenshäufigkeit als Schätzwert für die Übergangswahrscheinlichkeit.

11.2.2 Maximum-Likelihood-Schätzer für das Bi-Gramm-Modell

Nun soll der ML-Schätzwert für die Übergangswahrscheinlichkeiten des Bi-Gramm-Modells $p(v_k|v_j)$ mit $v_j, v_k \in V$ ermittelt werden. Dazu betrachten wir für ein festes Wort $v_j \in V$ den Zustandsraum $\tilde{V}_j = \{v_j\} \times V$, also alle möglichen Zweier-Wortkombinationen mit festem ersten Wort v_j . Für jedes $k \in \{1, \dots, s\}$ sei $Y_k : \Omega \rightarrow \mathbb{N}_0$ die Zufallsvariable, die die Anzahl der in n Versuchen erhaltenen Ausgänge (v_j, v_k) beschreibt. Das Bi-Gramm-Modell kann nun als Multinomialexperiment modelliert werden, welches n -mal in unabhängiger Folge durchgeführt wird. In einem theoretischen Modell liegen Wahrscheinlichkeiten $p(v_j, v_1), \dots, p(v_j, v_s)$ vor. Es gilt also $(Y_1, \dots, Y_s) \sim \text{Mult}(n; p(v_j, v_1), \dots, p(v_j, v_s))$. Dann folgt mit obiger Erkenntnis beim Ermitteln des ML-Schätzwerts für das Uni-Gramm-Modell, dass

$$\hat{p}(v_k|v_j) = \frac{c(v_j, v_k)}{\sum_{r=1}^s c(v_j, v_r)} \quad \forall k, j = 1, \dots, s \quad (11.11)$$

ML-Schätzwert für die Übergangswahrscheinlichkeit des Bi-Gramm-Modells zu den beobachteten Ausgängen c_1, \dots, c_s eines Trainingsdatensatzes mit $\sum_{r=1}^s c(v_j, v_r) = n$ für alle $j = 1, \dots, s$ ist. Dies entspricht der beobachteten relativen Auftretenshäufigkeit einer Zweier-Wortsequenz bezüglich aller Zweier-Wortsequenzen mit gleichem ersten Wort. Sie kann aus dem Trainingsdatensatz durch Zählen der Zweier-Wortsequenzen gleicher Wortgeschichte ermittelt werden.

Kernaspekte: Wird zusätzlich zu den im vorangegangenen Abschnitt gemachten Vereinfachungen die Annahme getroffen, dass die Trainingstexte den Sprachgebrauch ausreichend gut repräsentieren, dann stellen die relativen Häufigkeiten der Wortsequenzen im Trainingstext (ML-)Schätzwerte für die Übergangswahrscheinlichkeiten dar.

11.3 Kombination der Modelle

Das größte Problem des N -Gramm-Modells liegt in der Spärlichkeit der vorhandenen Daten, dabei ist wie in Abschnitt 11.2.1 dargelegt, die Größe des Testdatensatzes ausschlaggebend für die Genauigkeit der Schätzung der Übergangswahrscheinlichkeiten und damit für die Angemessenheit des Modells. Geht man davon aus, dass der deutsche Standardwortschatz 75.000 Wörter umfasst (Güll, 2010, S.

52), so existieren prinzipiell ohne die Berücksichtigung grammatischer Regeln $|\text{Per}_2^{75.000}(mW)| = 75.000^2$ Zweier-Wortsequenzen (auch *Bi-Gramme* genannt) und $|\text{Per}_3^{75.000}(mW)| = 75.000^3$ Dreier-Wortsequenzen (auch *Tri-Gramme* genannt). Nun ist klar, dass selbst ein Trainingstext mit mehreren Millionen Wörtern nur einen Bruchteil aller Bi- und Tri-Gramme und diese zum Teil auch nur mit einer geringen Häufigkeit enthalten wird. Viele Bi-Gramme sind somit gar nicht oder nur mit einer sehr geringen Häufigkeit im Trainingsdatensatz anzutreffen. Aber auch bei einzelnen Wörtern (auch *Uni-Gramme* genannt) kann es sein, dass diese nicht im Trainingsdatensatz enthalten sind. Insgesamt bestehen somit drei Probleme: das der nicht auftretenden, auch als *ungesehen* bezeichneten einzelnen Wörter, das der ungesehnen Wortsequenzen und das der geringen Auftretenshäufigkeiten der Wortsequenzen im Trainingsdatensatz.

Das Problem der geringen Auftretenshäufigkeiten ist aufgrund der deutlich größeren Anzahl an prinzipiell existierenden Tri-Grammen bei Tri-Gramm-Modellen umso größer (Ney et al., 1997, S. 176). Es wird hier sehr häufig passieren, dass die Wortvorgeschichte nicht im Trainingsdatensatz enthalten ist und die Schätzung der Übergangswahrscheinlichkeiten im Tri-Gramm-Modell daher häufig ungenau ist.

Ein Vergleich der Modelle zeigt, dass die Verlässlichkeit der Schätzung bei N -Gramm-Modellen höherer Ordnung aufgrund geringerer Zählhäufigkeiten abnimmt. Die Abhängigkeit von den vorangegangenen Wörtern und damit des Satzkontextes kann allerdings besser modelliert werden, je höher die Ordnung des N -Gramm-Modells gewählt wird. Nach Möglichkeit sollte die Schätzung der Übergangswahrscheinlichkeit beides sein – kontextbasiert und verlässlich. Häufig werden daher N -Gramm-Modelle unterschiedlicher Ordnung N kombiniert.

Die wichtigsten Methoden zur Kombination der Modelle lassen sich in zwei Kategorien unterteilen: Rückfall-Strategien und lineare Interpolation der Modelle (Chen & Goodman, 1998, S. 17 - 18). Beide Kombinationsmöglichkeiten sollen im Folgenden vorgestellt werden.

11.3.1 Rückfallstrategie

Bei der Nutzung des Sprachmodells zur Wortvorhersage möchte man aufgrund des hohen Kontextbezugs idealerweise das Modell höchster Ordnung, also das Tri-Gramm-Modell, verwenden. Es kann nun allerdings passieren, dass eine Wortvorgeschichte der Länge zwei nicht im Trainingsdatensatz aufgetreten ist. In diesem Fall ist die Auftretenshäufigkeit jedes Tri-Grammes, das mit dieser Wortvorgeschichte startet, null. Da die Übergangswahrscheinlichkeit im Tri-Gramm-Modell über die bedingte relative Auftretenshäufigkeit geschätzt wird, ist auch diese gleich null. Die Stichprobengröße ist somit zu klein, um die Übergangswahrscheinlichkeiten im Tri-Gramm-Modell sinnvoll schätzen zu können. Ebenso verhält es sich mit der Schätzung der Übergangswahrscheinlichkeiten im Bi-Gramm-Modell, wenn eine Wortvorgeschichte der Länge eins nicht im Trainingsdatensatz beobachtet wurde. Um einen Vorschlag

machen zu können, wird somit diejenige relative Häufigkeit gemäß der Funktion

$$h(w_i|w_{i-2}, w_{i-1}) := \begin{cases} \frac{c(w_{i-2}, w_{i-1}, w_i)}{\sum_{r=1}^s c(w_{i-2}, w_{i-1}, v_r)} & , c(w_{i-2}, w_{i-1}) > 0 \\ \frac{c(w_{i-1}, w_i)}{\sum_{r=1}^s c(w_{i-1}, v_r)} & , c(w_{i-1}) > 0 \text{ und } c(w_{i-2}, w_{i-1}) = 0 \\ \frac{c(w_i)}{\sum_{r=1}^s c(v_r)} & , \text{sonst} \end{cases}$$

als Schätzwert für die Übergangswahrscheinlichkeit verwendet.

Bei dieser Rückfallstrategie kann es passieren, dass die relative Häufigkeit der Wortsequenz w_{i-2}, w_{i-1}, w_i maßgeblich für die Wortvorhersage ist, obwohl sie auf einer geringen beobachteten Auftretenshäufigkeit basiert. Daher gibt es Methoden, bei denen die Zählhäufigkeit der N -Gramme bei der Entscheidung, wie lange die betrachtete Wortvorgeschichte gewählt werden soll, berücksichtigt wird (Chen & Goodman, 1998, S.11 - 12). Diese werden allerdings schnell komplex, weshalb stattdessen mit Blick auf die Zugänglichkeit für Lernende eine einfachere Methode zur Kombination der Modelle vorgestellt wird.

11.3.2 Interpolation

Wenn nicht genügend Daten vorhanden sind, um eine Wahrscheinlichkeit im Modell höherer Ordnung sinnvoll zu schätzen, kann das Modell niedrigerer Ordnung oft nützliche Informationen liefern. Bei dieser Methode werden daher N -Gramm-Modelle unterschiedlicher Ordnung linear interpoliert, und die Verteilung des interpolierten Tri-Gramm-Modells ist eine Mischverteilung der Verteilungen der einzelnen N -Gramm-Modelle. Die Übergangswahrscheinlichkeit ergibt sich dann als gewichtete Summe der einzelnen Uni-, Bi-, und Tri-Gramm-Wahrscheinlichkeiten zu

$$p_{int}^*(w_i|w_{i-2}, w_{i-1}) := g_1 p(w_i) + g_2 p(w_i|w_{i-1}) + g_3 p(w_i|w_{i-2}, w_{i-1}), \quad i \geq 3,$$

mit $g_1, g_2, g_3 \geq 0$ und $\sum_{j=1}^3 g_j = 1$ (Chen & Goodman, 1998, S. 10; Fink, 2003, S. 109).

Die Tatsache, dass Wörtern, die nicht im Trainingsdatensatz vorkommen, die Wahrscheinlichkeit null zugewiesen wird, obwohl sie im Sprachgebrauch natürlich verwendet werden, wird bislang nicht berücksichtigt. Um dieses Problem zu beheben, wird die bisherige Verteilung mit einer Verteilung interpoliert, die jedem Wort im Vokabular eine positive Wahrscheinlichkeit zuweist. Die einfachste Möglichkeit ist die Interpolation mit einer Gleichverteilung auf allen Wörtern des Vokabulars (Chen & Goodman, 1998, S. 10). Zur Übergangswahrscheinlichkeit $p_{int}^*(w_i|w_{i-2}, w_{i-1})$ wird dann die Glättungswahrscheinlichkeit $\frac{1}{|V|}$ addiert. Die Wahrscheinlichkeit für das nächste Wort wird dann über

$$p_{int}(w_i|w_{i-2}, w_{i-1}) := g_0 \frac{1}{|V|} + g_1 p(w_i) + g_2 p(w_i|w_{i-1}) + g_3 p(w_i|w_{i-2}, w_{i-1}), \quad i \geq 2$$

mit $g_0, g_1, g_2, g_3 \geq 0$ und $\sum_{j=0}^3 g_j = 1$ bestimmt (Chen & Goodman, 1998, S. 10; Fink, 2003, S. 109). Diese Art der Interpolation bezeichnet man auch als Jelinek-Mercer Glättung (engl. Jelinek-Mercer Smoothing) (Chen & Goodman, 1998, S. 10).

Für die Anwendung des Sprachmodells zur Wortvorhersage spielt die Berücksichtigung ungewohnter Wörter eine untergeordnete Rolle, da ohnehin nur die Wörter mit den höchsten Übergangswahrscheinlichkeiten betrachtet werden. In Abschnitt 11.4 wird jedoch deutlich, dass die Berücksichtigung ungewohnter Wörter für die Bewertung des Sprachmodells, die unabhängig von dessen spezifischem Anwendungsbereich erfolgt, von großer Bedeutung ist. Im Lernmaterial wird dieser Aspekt daher erst im Kontext der Modellbewertung eingeführt (vgl. Abschnitt 13.2.5).

In obiger Definition ist die interpolierte Übergangswahrscheinlichkeit bisher lediglich für $i \geq 2$ definiert. Damit nicht unterschieden werden muss, welche Übergangswahrscheinlichkeit am Satzanfang genutzt wird, werden auch die Fälle $i = 1$ und $i = 2$ einbezogen und

$$p_{int}(w_i | w_{i-2}, w_{i-1}) := \begin{cases} g_0 \cdot \frac{1}{|V|} + g_1 p(w_i) & , i = 1, \\ \frac{g_0}{|V|} + g_1 p(w_i) + g_2 p(w_i | w_{i-1}) & , i = 2, \\ \frac{g_0}{|V|} + g_1 p(w_i) + g_2 p(w_i | w_{i-1}) + g_3 p(w_i | w_{i-2}, w_{i-1}) & , \text{sonst}, \end{cases}$$

mit $g_0, g_1, g_2, g_3 \geq 0$ und $\sum_{j=0}^3 g_j = 1$ definiert. Die gemeinsame Verteilung der Zufallsvariablen X_1, \dots, X_n ist dann mit (11.4) über

$$P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_m = w_m) := \prod_{i=1}^m p_{int}(w_i | w_{i-2}, w_{i-1}) \quad (11.12)$$

mit $g_0, g_1, g_2, g_3 \geq 0$ und $\sum_{j=0}^3 g_j = 1$ gegeben.

Alle einzelnen N -Gramm-Modelle können ebenfalls als interpolierte Sprachmodelle dargestellt werden, indem bestimmte Interpolationsgewichte auf null gesetzt werden. Wenn im Folgenden von einem *geglätteten* Sprachmodell die Rede ist, bezieht sich dies stets auf ein N -Gramm-Modell, dessen Wahrscheinlichkeitsverteilung mit einer Gleichverteilung interpoliert wurde, und es gilt $g_0 > 0$. Die Interpolationsgewichte für das kombinierte N -Gramm-Modell können zunächst auf Basis heuristischer Überlegungen festgelegt werden. Später wird gezeigt, wie auch diese als Parameter der Verteilung in (11.12) geschätzt werden können.

Kernaspekte: Um die Vorteile aller Modelle zu nutzen, können diese entweder über eine Rückfallstrategie oder eine Interpolation kombiniert werden. Bei der Rückfallstrategie werden Vorhersagen anhand des N -Gramm-Modells höchster Ordnung getroffen, bei welchem die Länge der Wortvorgeschichte kurz genug ist, sodass deren Auftretenshäufigkeit größer als null ist. Bei der Rückfallstrategie wird die Vorhersage basierend auf der Übergangswahrscheinlichkeit desjenigen N -Gramm-Modells mit der höchsten Ordnung und bei welchem die Auftretenshäufigkeit der Wortvorgeschichte ungleich null ist, gemacht. Bei der Interpolation der Modelle ergibt sich die Übergangswahrscheinlichkeit als gewichteter Mittelwert der Übergangswahrscheinlichkeiten der einzelnen Modelle.

11.4 Bewertung

Ist nun nach der *Trainingsphase* ein erstes Modell entwickelt worden, so drängt sich die Frage auf, wie gut das Modell die zugrunde liegenden Daten beschreiben und somit die Muster in vorliegenden Texten widerspiegeln kann. In der *Testphase* soll das Sprachmodell daher evaluiert werden, bevor es in der *Anwendungsphase* tatsächlich eingesetzt wird. Als Maß für die Qualität eines Sprachmodells hat sich die *Kreuzentropie* etabliert (Carstensen et al., 2009, S. 130; Chen & Goodman, 1998, S. 7). Als statistisches Modell gibt das N -Gramm-Modell lediglich eine Näherung für die tatsächliche Wahrscheinlichkeitsverteilung an. Die Kreuzentropie misst die Unterschiedlichkeit der wahren Verteilung und derjenigen Verteilung des Modells und damit die Qualität der Näherung. Sie ist wie folgt definiert:

Sei Z mit $Z : \Omega \rightarrow V$ eine Zufallsvariable über einem Grundraum Ω mit der diskreten Verteilung q , und sei p eine weitere diskrete Verteilung auf V (meistens die Verteilung des Modells). Dann heißt

$$H_q(p) := -\mathbb{E}_q[\log p(Z)] = -\sum_{w \in V} q(Z = w) \log p(Z = w)$$

Kreuzentropie von Z bezüglich p (Carstensen et al., 2009, S. 127; Manning & Schütze, 1999, S. 74 - 75).

Aufgrund der Tatsache, dass die wahre Verteilung q üblicherweise nicht vorliegt, kann nur die sogenannte *geschätzte Kreuzentropie* oder auch *genäherte Kreuzentropie* bestimmt werden. Seien hierzu Z_1, \dots, Z_n mit $Z_i : \Omega \rightarrow V$ Zufallsvariablen über einem Grundraum Ω mit der diskreten Verteilung q , und sei p eine weitere diskrete Verteilung auf V . Die wahre Verteilung wird bei der genäherten Kreuzentropie mithilfe der *Testdaten*, die als Zufallsstichprobe aus der wahren Verteilung q dient, näherungsweise dargestellt. Die genäherte Kreuzentropie der Verteilung p des Modells relativ zu einem Testdatensatz $T = \{w_1, \dots, w_n\}$ ist dann definiert als

$$K_T(p) := -\frac{1}{n} \log p(Z_1 = w_1, \dots, Z_n = w_n)$$

(Manning & Schütze, 1999, S. 75). Wird im Folgenden von der Kreuzentropie gesprochen, so ist damit immer die genäherte Kreuzentropie gemeint. Je kleiner der Wert der Kreuzentropie, desto besser beschreibt die Verteilung des Modells die wahre Verteilung genähert durch den vorliegenden Testdatensatz.

Für die Verteilung des interpolierten Tri-Gramm-Modells P_{g_0, g_1, g_2, g_3} lässt sich die Kreuzentropie mit (11.12) dann zu

$$\begin{aligned} K_T(P_{g_0, g_1, g_2, g_3}) &= -\frac{1}{n} \log P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_n = w_n) \\ &= -\frac{1}{n} \sum_{i=1}^n \log p_{int}(w_i | w_{i-2}, w_{i-1}) \end{aligned} \tag{11.13}$$

bestimmen. Daraus wird ersichtlich, warum bereits ein einziges Wort im Testdatensatz, das im Trainingsdatensatz nicht vorkommt, erhebliche Auswirkungen auf die

Bewertung eines nicht geglätteten Sprachmodells hat: Eine einzelne Übergangswahrscheinlichkeit, die null ist, führt dazu, dass $P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_n = w_n) = 0$ gilt, wodurch die Kreuzentropie unendlich groß wird.

Die Wahl dieses Qualitätsmaßes ist auch über das Maximum-Likelihood-Prinzip motivierbar, schließlich ist die genäherte Kreuzentropie eng mit der Log-Likelihood-Funktion der Verteilung P_{g_0, g_1, g_2, g_3} zum beobachteten Testdatensatz verbunden: Sie entspricht der mit dem Faktor $-\frac{1}{n}$ skalierten Log-Likelihood-Funktion zum Testdatensatz. Nach dem Maximum-Likelihood-Prinzip wird demjenigen Modell eine große Glaubwürdigkeit zugesprochen, unter welchem der Testdatensatz eine große Auftretenswahrscheinlichkeit besitzt, der Wert der Likelihood-Funktion sowie der Log-Likelihood-Funktion also hoch ist. Dies entspricht einer kleinen Kreuzentropie. Modellen mit kleiner Kreuzentropie eine hohe Qualität zuzuordnen erweist sich somit auch aus dieser Perspektive als sinnhaft. Die Skalierung mit der Größe des Testdatensatzes sorgt dafür, dass bei einem Vergleich mit Sprachmodellen, die auf anderen Testdatensätzen getestet werden, keine Verzerrung entsteht. So wird verhindert, dass Modelle, die auf kleineren Testdatensätzen getestet werden, bei denen die Likelihood somit typischerweise geringer ausfällt, einen unberechtigten Vorteil erhalten.

Auch aus informationstheoretischer Sicht ist die Sinnhaftigkeit dieses Qualitätsmaßes erklärbar. Vereinfachend dargestellt wird $-\log_2 p_{int}(w_i | w_{i-2}, w_{i-1})$ hier als Informationsgehalt oder auch als Überraschung des Ereignisses „das Wort w_i tritt nach der Sequenz w_{i-2}, w_{i-1} auf“ bezeichnet. Ziel eines Sprachmodells ist es, die Überraschung möglichst gering zu halten. Es wird nun für jedes Wort im Testdatensatz überprüft, wie hoch die Überraschung ausfällt. Ist sie klein, so scheint das Sprachmodell die Wörter im Testdatensatz gut vorhersagen zu können. Für eine ausführlichere und fundiertere Erklärung sei an dieser Stelle auf Carstensen et al. (2009, S. 126 - 130) sowie Manning und Schütze (1999, S. 73 - 76) verwiesen. Da die Definition der Kreuzentropie ursprünglich aus der Informationstheorie stammt, wird statt des natürlichen Logarithmus in der Regel der Logarithmus zur Basis zwei verwendet. Die Kreuzentropie und die Entropie, aus der die Kreuzentropie abgeleitet wird, können allerdings mit beliebigen Logarithmusbasen definiert werden, worauf zum Beispiel in Cover und Thomas (1991, S. 13 - 14) explizit hingewiesen wird. In Anbetracht dessen, dass Schülerinnen und Schüler aus dem Unterricht nur den natürlichen Logarithmus kennen, wird im Lernmaterial der Logarithmus zur Basis e verwendet (vgl. Abschnitt 13.2.5).

Die Kreuzentropie eignet sich auch als Zielfunktion eines Optimierungsproblems, um bestmögliche Interpolationsgewichte zu finden. Sie wird in diesem Zusammenhang dann auch als Kreuzentropie-Fehlerfunktion (engl. cross entropy loss) bezeichnet und beispielsweise auch bei anderen ML-Methoden wie der Logistischen Regression genutzt, um Parameter zu schätzen (Bishop, 2007, S. 206). Im nächsten Abschnitt soll dieser Ansatz verfolgt werden.

Kernaspekte: Als Qualitätsmaß hat sich die Kreuzentropie etabliert. Sie entspricht der mit dem Faktor $-\frac{1}{n}$ skalierten Log-Likelihood-Funktion zum Testdatensatz.

11.5 Schätzen der Interpolationsgewichte

Auch für das Schätzen der Interpolationsgewichte als Parameter einer Verteilungsfunktion zu gegebenen Daten kann das Maximum-Likelihood-Prinzip herangezogen werden. Angenommen, die Übergangswahrscheinlichkeiten sind bekannt und ein Trainingsdatensatz w_1, \dots, w_n liegt vor, so lässt sich der ML-Schätzwert als Maximum der Log-Likelihood-Funktion

$$\log \mathcal{L}_{c_1, \dots, c_s}(g_0, g_1, g_2, g_3) := \log P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_n = w_n)$$

unter der Nebenbedingung $g_j \geq 0$ für alle j und $\sum_j g_j = 1$ bestimmen (Ney et al., 1997, S. 195; Wandmacher, 2008, S. 50). Prinzipiell könnte man die Nebenbedingung auf $g_j > 0$ verschärfen, da anhand der Überlegungen in Abschnitt 11.3 klar ist, dass alle einzelnen Verteilungen mit einbezogen werden sollen. Diese Verschärfung wird hier jedoch nicht vorgenommen, um, wie im weiteren Verlauf deutlich wird, die eindeutige Lösbarkeit des Optimierungsproblems zu gewährleisten.

Gemäß der Beziehung (11.13) entspricht dies dem Minimieren der Kreuzentropie in Abhängigkeit der Gewichte. Das resultierende Optimierungsproblem kann wie folgt formuliert werden: Minimiere die Zielfunktion

$$\widetilde{K}(g_0, g_1, g_2, g_3) := K_T(P_{g_0, g_1, g_2, g_3}) = -\frac{1}{n} \sum_{i=1}^n \log p_{int}(w_i | w_{i-2}, w_{i-1})$$

auf der Menge $M = \{(g_0, g_1, g_2, g_3) \in \mathbb{R}^4 : g_0, g_1, g_2, g_3 \geq 0, \sum_{j=0}^3 g_j = 1\}$. Hierbei handelt es sich um ein mehrdimensionales nichtlineares Optimierungsproblem unter Nebenbedingungen. Es ist offensichtlich, dass der Faktor $-\frac{1}{n}$ für die Optimierung keinen Einfluss hat. Da jedoch im Lernmaterial die Kreuzentropie anstelle der Likelihood-Funktion verwendet wird, wird hier ebenfalls die Kreuzentropie als Zielfunktion beibehalten, um die Konsistenz in der Darstellung zu gewährleisten.

11.5.1 Existenz und Eindeutigkeit der Lösung des Optimierungsproblems

Um die Eignung der Kreuzentropie als Zielfunktion für das Ermitteln der optimalen Gewichte zu untersuchen, werden Überlegungen zur Lösbarkeit des obigen Optimierungsproblems angestellt. Die Existenz und Eindeutigkeit der Lösung wird im Lernmaterial nicht thematisiert. Da die Lernenden später allerdings ein Optimierungsverfahren entwickeln sollen, sollte die Lösbarkeit auch aus diesem Grund vorab untersucht werden, sodass die Ergebnisse der Lernenden korrekt eingeschätzt werden können. Falls das Problem keine eindeutige oder gar keine Lösung besitzt, kann das Optimierungsverfahren entweder zu unterschiedlichen Ergebnissen oder zu falschen Lösungen führen. Im Folgenden kann allerdings gezeigt werden, dass das vorliegende Optimierungsproblem eindeutig lösbar ist.

Da die Menge M beschränkt und abgeschlossen ist und die Zielfunktion zudem stetig ist, existiert eine Lösung des obigen Optimierungsproblems. Um die eindeutige

Lösbarkeit des Optimierungsproblems zu zeigen, wird die Funktion

$$f(g_0, g_1, g_2, g_3) := -\log(g_0 p_0 + g_1 p_1 + g_2 p_2 + g_3 p_3)$$

mit $f : M \rightarrow \mathbb{R}$ und $p_0 := \frac{1}{|V|}, p_1 := p(w_i), p_2 := p(w_i | w_{i-1}), p_3 := p(w_i | w_{i-2}, w_{i-1})$ definiert. Die Funktion f ist strikt konvex wegen

$$\begin{aligned} f(\Theta x_0 + (1 - \Theta)y_0, \Theta x_1 + (1 - \Theta)y_1, \Theta x_2 + (1 - \Theta)y_2) \\ = -\log \left(\sum_{j=0}^3 (\Theta x_j + (1 - \Theta)y_j) p_j \right) \\ = -\log \left(\sum_{j=0}^3 \Theta x_j p_j + (1 - \Theta) y_j p_j \right) \\ < -\Theta \log \sum_{j=0}^3 x_j p_j - (1 - \Theta) \log \sum_{j=0}^3 y_j p_j \\ = \Theta f(x_0, x_1, x_2, x_3) + (1 - \Theta) f(y_0, y_1, y_2, y_3) \end{aligned}$$

für alle $(x_0, x_1, x_2, x_3), (y_0, y_1, y_2, y_3) \in M$ und alle $\Theta \in (0, 1)$. Dabei ist in die Abschätzung die Jensensche Ungleichung und, dass für alle $(x_0, x_1, x_2, x_3) \neq (y_0, y_1, y_2, y_3)$ im Allgemeinen $\sum_{j=0}^3 x_j p_j \neq \sum_{j=0}^3 y_j p_j$ folgt, eingegangen. Die Jensensche Ungleichung besagt, dass für eine strikt konvexe Funktion g (wie hier dem negativen Logarithmus)

$$g \left(\sum_{i=1}^n \gamma_i x_i \right) < \sum_{i=1}^n \gamma_i g(x_i)$$

für alle $\gamma_i > 0$ mit $\sum_i \gamma_i = 1$ gilt, sofern nicht alle x_i gleich sind (Pečarić et al., 2015, S. 3). Die Zielfunktion $\widetilde{K}(g_0, g_1, g_2, g_3)$ ist dann als Summe strikt konvexer Funktionen wieder eine strikt konvexe Funktion und das Optimierungsproblem auf der konvexen Menge M eindeutig lösbar.

11.5.2 Lösen des Optimierungsproblems

Für eine einfachere Handhabbarkeit des Optimierungsproblems kann dieses auch als Minimierungsproblem formuliert werden, bei welchem die Zielfunktion

$$\widetilde{K}^*(g_0, g_1, g_2) := \widetilde{K}(g_0, g_1, g_2, 1 - g_0 - g_1 - g_2)$$

auf $M^* = \{(g_0, g_1, g_2) \in \mathbb{R}^3 : 0 \leq g_0, g_1, g_2 \leq 1\}$ minimiert wird. Obige Aussagen für die Existenz und Eindeutigkeit des Optimierungsproblems gelten dann immer noch, wie leicht einzusehen ist.

In der Praxis wird das Optimierungsproblem meist mit dem EM-Algorithmus gelöst, der die Gewichte in einem iterativen Verfahren bestimmt (Ney et al., 1997, S. 195; Wandmacher, 2008, S. 50). Dieser Algorithmus zählt zu den Greedy-Algorithmen

(Wandmacher, 2008, S. 50). Das bedeutet auch, dass der Algorithmus möglicherweise nur ein lokales Minimum findet. Mit Lernenden kann hier entweder lediglich das Optimierungsproblem formuliert werden, oder alternativ ist auch eine vollständige Suche auf dem diskretisierten Suchraum M^* möglich. Hierbei werden systematisch alle möglichen Kombinationen der Gewichtung im diskretisierten Suchraum durchprobiert, deren Funktionswert $\tilde{K}^*(g_0, g_1, g_2)$ berechnet und mit dem bisher niedrigsten Funktionswert verglichen. Ein Nachteil der vollständigen Suche ist der hohe Rechenaufwand. Dafür ist die Methode einfach zugänglich und findet innerhalb der gewählten Diskretisierung immer das globale Optimum (Wandmacher, 2008, S. 50).

Diejenige Kombination der Interpolationsgewichte, die über einen Optimierungsalgorithmus gefunden wird und die Kreuzentropie minimiert, ist dann ein Schätzwert zum vorliegenden Trainingsdatensatz. Der Trainingsdatensatz hat unter dem interpolierten Sprachmodell mit diesen Gewichten die größte Auftretenshäufigkeit. Dabei ist zu beachten, dass der Trainingsdatensatz, mit dem die Übergangswahrscheinlichkeiten geschätzt werden, nicht identisch mit dem Datensatz sein darf, der zur Schätzung der Interpolationsgewichte verwendet wird. Andernfalls würde die Gewichtung mit $g_3 = 1$ und $g_0 = g_1 = g_2 = 0$ die Lösung des Optimierungsproblems ergeben (Chen & Goodman, 1998, S. 11).

Insgesamt kann so ein Sprachmodell entwickelt werden, dessen Parameter in einem zweistufigen Verfahren geschätzt werden: Zuerst werden die Übergangswahrscheinlichkeiten mithilfe eines ersten Trainingsdatensatzes geschätzt, und anschließend erfolgt die Schätzung der Interpolationsgewichte mithilfe eines zweiten Trainingsdatensatzes.

Kernaspekte: Die Interpolationsgewichte können über das Maximum-Likelihood-Prinzip geschätzt werden, was äquivalent zum Minimieren der Kreuzentropie in Abhängigkeit der Gewichte ist. Das zugehörige Optimierungsproblem ist eindeutig lösbar, und die Lösung kann zum Beispiel über eine vollständige Suche bestimmt werden.

11.6 Stärken und Schwächen des N -Gramm-Modells

Die vorangegangenen Abschnitte zeigen, dass N -Gramm-Modelle auf mehreren stark vereinfachenden Annahmen beruhen. Dass diese auch Auswirkungen auf die Qualität der Sprachmodellierung haben ist selbstverständlich. Im Folgenden sollen diese Schwächen, aber auch die Stärken von N -Gramm-Modellen zusammengetragen werden.

N -Gramm-Modelle sind insbesondere aufgrund ihrer einfachen Funktionsweise geschätzt. Im Gegensatz zu anderen Modellen ist das N -Gramm-Modell weniger komplex und lässt sich einfach in größere Systeme wie zum Beispiel als Sprachmodell in Spracherkennungssystemen integrieren.

Ein zentraler Vorteil ist außerdem die Flexibilität der Modelle: Sie können ohne zusätzliche Sprachkenntnisse allein durch geeignete Trainingsdaten für jede beliebige

Sprache und Textart adaptiert werden (Pfister & Kaufmann, 2017, S. 386). Das Sprachmodell ließe sich dadurch auch leicht auf den individuellen Wortschatz der Nutzenden abstimmen.

Allerdings ist die Qualität der Trainingsdaten entscheidend für die Leistungsfähigkeit des Modells. Nur wenn die Trainingsdaten hinsichtlich Vokabular und Textart auf den jeweiligen Anwendungskontext abgestimmt sind, können passende Vorschläge für das nächste Wort gemacht werden. Das Zusammenstellen geeigneter Trainingsdatensätze kann in der Praxis jedoch viel Aufwand erfordern (Pfister & Kaufmann, 2017, S. 387).

Eine weitere Schwäche liegt in der fehlenden Berücksichtigung grammatischer Regeln. *N*-Gramm-Modelle basieren auf Statistiken von Textdaten. Das ist vor allem bei sogenannten stark flektierten Sprachen, also Sprachen in denen Verben, Nomen und Adjektive viele verschiedene Formen haben, ein Problem (Pfister & Kaufmann, 2017, S. 387). Jede Form eines Wortes wird als neues Uni-Gramm gesehen, anstatt die unterschiedlichen Wortformen einem Wortstamm zuzuordnen und später zu konjugieren. Zudem filtert das System grammatisch falsche Wortvorschläge nicht heraus, da die zugrunde liegenden *N*-Gramm-Modelle in der Praxis üblicherweise mit kleinen Ordnungen *N* arbeiten.

Außerdem können Abhängigkeiten nur über wenige Wörter hinweg und nicht über mehrere Wortsequenzen oder gar Sätze übergreifend erfasst werden. Die Reduktion der Länge der betrachteten Wortsequenzen stellt einen Kompromiss dar, der zugunsten der Reduktion der Modellkomplexität in Kauf genommen werden muss. Je länger die analysierten Wortfolgen sind, desto seltener treten sie in den Trainingsdaten auf. Dies führt zu einer abnehmenden Verlässlichkeit der Schätzung der Modellparameter. Um Kontext umfassender zu berücksichtigen, werden in der Regel Transformer Modelle eingesetzt, die die Beziehungen zwischen Wörtern in einem Text, unabhängig von deren Abstand zueinander erkennen (Vaswani et al., 2017, S. 3 - 5).

Zudem geht man bei der Modellierung der Sprache mit den beschriebenen statistischen Methoden davon aus, dass die Sprache selbst eine feste Größe ist, und dass die statistischen Modelle, sowie deren Parameter nur möglichst gut gewählt werden müssen. Sprache befindet sich tatsächlich aber in stetigem Wandel. Das wird bei der deutschen Sprache insbesondere deutlich, wenn die deutschen Jugendwörter der vergangenen Jahre in den Blick genommen werden. Hier ist die hohe Anzahl der Anglizismen wie „slay“ oder „cringe“ auffällig (NDR, 2024), während andere Worte aus dem Sprachgebrauch verschwinden.

12 Anwendung der mathematischen Methoden mit realen Textdaten

Der im Lernmaterial verwendete Datensatz umfasst ca. 350.000 Wörter. Er setzt sich aus den deutschsprachigen Texten des Korpus „What’s up, Switzerland“ (Stark et al., 2020) und den Texten der Kategorie „Belletristik“ des Korpus „LIMAS“ (Forschungsgruppe LIMAS, 1971) zusammen. Der Korpus „LIMAS“ ist ein Zeitschnittkorpus der deutschen Gegenwartssprache von 1970, der öffentlich zugänglich ist. Der Korpus „What’s up, Switzerland“ besteht aus verschiedenen Subkorpora, welche im Jahr 2014 gesammelte anonymisierte Chatnachrichten in verschiedenen in der Schweiz gängigen Sprachen enthalten. Für dieses Lernmaterial wurde der Subkorpus der Sprache nichtdialektisches Deutsch ausgewählt. Die sogenannte *Tokenisierung*, also die Zerlegung des Datensatzes in einzelne Wörter und Abspeicherung in listenähnlichen Strukturen, wird ausgelagert aus dem Lernmaterial vorgenommen, da es sich hier um eine rein technische Aufgaben handelt. Anschließend wird eine Teilung des Datensatzes in Test- und Trainingsdaten im Verhältnis eins zu vier vorgenommen.

Es werden unterschiedliche Modelle entwickelt, getestet und anhand der Kreuzentropie bewertet. Dazu zählen das geglättete Uni-Gramm-Modell, das geglättete Bi-Gramm-Modell, das geglättete Tri-Gramm-Modell sowie das in Abschnitt 11.3 vorgestellte interpolierte Tri-Gramm-Modell. Zusätzlich wird ein als *Zufalls-Modell* bezeichnetes Modell getestet, bei welchem eine Gleichverteilung auf allen Wörtern des Vokabulars angenommen wird. Als Größe des Vokabulars wird dabei 75.000 verwendet, was der geschätzten Anzahl der Wörter des deutschen Standardwortschatzes entspricht (Güll, 2010, S. 52). Tabelle 12.1 stellt die über Gleichung (11.13) ermittelte Kreuzentropie für die einzelnen Modelle sowie deren Interpolationsgewichte dar¹. Es ist ersichtlich, dass die Kreuzentropie des Zufalls-Modells am größten ist. Schlussfolgernd daraus kann davon ausgegangen werden, dass dieses Modell den Sprachgebrauch für diese Art von Texten am schlechtesten modelliert, was angesichts der unzulänglichen Modellierungsannahmen zu erwarten war. Das geglättete Uni-, Bi- und Tri-Gramm-Modell schneiden besser ab. Besonders auffällig ist hier, dass sich die Kreuzentropie zwischen Uni- und Bi-Gramm-Modell wesentlich mehr unterscheidet als die zwischen Bi- und Tri-Gramm-Modell. Die Kreuzentropie des geglätteten interpolierten Tri-Gramm-Modells mit gleich großen Gewichten ist am

¹Da das Satzende nicht als Token erfasst wurde, die Größe der Datensätze im Vergleich zu denjenigen bei prädiktiven Textsystemen typischen gering ist, sowie aus didaktischen Gründen eine andere Basis für den Logarithmus bei der Berechnung der Kreuzentropie verwendet wurde, dienen die Bewertungsergebnisse primär zum Vergleich der Modelle untereinander und sind nicht direkt mit Werten aus der Literatur vergleichbar.

Tabelle 12.1: Kreuzentropie verschiedener Modelle (alle Werte sind auf zwei Stellen gerundet)

Modell	Interpolationsgewichte				Kreuzentropie
	g_0	g_1	g_2	g_3	
Zufalls-Modell	1	0	0	0	11,23
geglättetes Uni-Gramm-Modell	0,1	0,9	0	0	7,89
geglättetes Bi-Gramm-Modell	0,1	0	0,9	0	5,96
geglättetes Tri-Gramm Modell	0,1	0	0	0,9	5,34
geglättetes, interpoliertes Tri-Gramm-Modell	0,25	0,25	0,25	0,25	4,05
geglättetes, interpoliertes Tri-Gramm-Modell	0,04	0,20	0,14	0,62	3,76

kleinsten. Das Sprachmodell, welches N -Gramm-Modelle aller drei Ordnungen einbezieht, modelliert den Sprachgebrauch für diese Art von Texten somit am besten. Die numerische Ergebnisse bestätigen damit die theoretischen Überlegungen aus Abschnitt 11.3.

Beim interpolierten Tri-Gramm-Modell wurden zunächst gleich große Interpolationsgewichte gewählt. Über ein Optimierungsverfahren können anschließend diejenigen Gewichte bestimmt werden, welche die Kreuzentropie für das Modell zum gegebenen Testdatensatz minimieren. Hierzu wurde die Bibliothek Ipopt (Wächter & Biegler, 2006) verwendet. Es werden die in der letzten Zeile von Tabelle 12.1 aufgeführten Gewichte als Lösung des Optimierungsproblems zu einer Kreuzentropie von 3,76 ermittelt. Das Ergebnis für die Gewichtung ist mit dem anderer Forschungsarbeiten vergleichbar (Haque et al., 2016, S. 71).

Um die Qualität des interpolierten Modells mit optimierter Gewichtung mit den weiteren in Tabelle 12.1 aufgeführten Sprachmodellen zu vergleichen, müsste es jedoch auf neuen Daten getestet und die Kreuzentropie für diesen Datensatz berechnet werden. Der Ausgangsdatensatz müsste somit in drei Datensätze – zwei Trainingsdatensätze und ein Testdatensatz – unterteilt werden. Im Lernmaterial wird ausdrücklich darauf hingewiesen, dass der ermittelte Kreuzentropiewert des optimierten Modells lediglich eine erste Orientierung bietet, aber keine konkrete Aussage über die tatsächliche Modellleistung erlaubt. Aufgrund des Zeitaufwandes im Vergleich zum Lernzuwachs wird allerdings bewusst auf eine erneute Bewertung verzichtet und daher zu Beginn auch keine Dreiteilung des Datensatzes vorgenommen.

Zusätzlich wird ein zweiter Trainingsdatensatz bestehend aus Werken von Johann Wolfgang von Goethe (1774a, 1774b, 1796, 1809) mit rund 150.000 Wörtern to-

Tabelle 12.2: Kreuzentropie des geglätteten interpolierten Tri-Gramm-Modells mit Interpolationsgewichten $\tilde{g}_0, \tilde{g}_1, \tilde{g}_2, \tilde{g}_3 = 0, 25$ trainiert mit verschiedenen Datensätzen (alle Werte sind auf zwei Stellen gerundet)

Datensatz	Kreuzentropie
Datensatz LIMAS und Whats'up, Switzerland	4,05
Datensatz Goethe	8,34

kenisiert und im Lernmaterial in einem Zusatzblatt verwendet. Der Testdatensatz auf dem das mit den Texten von Goethe trainierte Modell getestet wird, bleibt jedoch unverändert. Tabelle 12.2 zeigt, dass dieses Sprachmodell eine deutlich höhere Kreuzentropie als das mit dem ursprünglichen Trainingsdatensatz trainierte Sprachmodell aufweist. Wie aufgrund der Unterschiedlichkeit von Test- und Trainingsdatensatz insbesondere hinsichtlich des Wortschatzes zu erwarten war, kann das mit dem Datensatz von Goethe trainierte Sprachmodell die zugrundeliegenden Testdaten weniger gut beschreiben.

13 Umsetzung im Lernmaterial

In diesem Kapitel wird gezeigt, wie mit Lernenden ein Sprachmodell zur Wortvorhersage, basierend auf Methoden aus der Schulmathematik entwickelt werden kann. Dazu werden zu den in Kapitel 11 vorgestellten mathematischen Inhalten anleitende Aufgaben formuliert und diese zu einer kohärenten Lernumgebung zusammengeführt. Da die didaktische Reduktion auf den in Kapitel 11 identifizierten Inhalten basiert, werden die bereits dort aufgeführten Quellenverweise hier nicht erneut genannt.

Tabelle 13.1 gibt eine Übersicht über den Aufbau und die Inhalte des Lernmaterials im Rahmen eines Modellierungstages. Im Lernmaterial wird zunächst das Bi-Gramm-Modell thematisiert indem erste intuitive Ideen gesammelt und mathematisiert werden. Das Modell wird schließlich erprobt und seine Grenzen werden erkundet. Das Uni- und Tri-Gramm-Modell werden im Anschluss als alternative Modelle eingeführt um verschiedenen Problemen, die beim Bi-Gramm-Modell aufgetreten sind, zu begegnen. Schließlich werden die einzelnen N -Gramm-Modelle interpoliert um die Vorteile aller einzelnen Modelle nutzen zu können. Zuletzt erfolgt die Bewertung des entwickelten Sprachmodells mit der Kreuzentropie. Diese wird im weiteren Verlauf genutzt, um die Interpolationsgewichte des interpolierten Tri-Gramm-Modells optimal festzulegen.

Ein besonderer Vorteil des Lernmaterials ist seine flexible Gestaltung: Der Workshop kann an verschiedenen Stellen verkürzt werden. Er lässt sich bereits nach Arbeitsblatt 1 oder 3 abschließen – entweder ohne Modellverbesserungen (vgl. Abschnitt 13.2.8) oder nach der Entwicklung der Modellverbesserungen. In beiden Fällen findet statt einer systematischen Bewertung der entwickelten Modelle lediglich eine erste Einschätzung auf Basis einzelner Beobachtungen statt. Dennoch wird der Modellierungskreislauf (mindestens) einmal vollständig durchlaufen. Abschließend folgt in beiden Szenarien eine Sicherung des vorangegangenen Arbeitsblattes, eine Sammlung der Grenzen des Modells sowie eine Diskussion zu problematischen Aspekten bei der Nutzung und Entwicklung von Sprachmodellen.

Das Lernmaterial baut auf zentralen Inhalten bis Jahrgangsstufe 9 auf und ist daher für Schülerinnen und Schüler ab Jahrgangsstufe 10 konzipiert. Es ist in Form von digitalen Arbeitsblättern, ergänzenden Tipp-, Hinweis-Karten und Zusatzaufgaben, sowie begleitenden Dokumentationsblättern und Präsentationsfolien umgesetzt. Bildschirmfotos des Lernmaterials sowie die Präsentationsfolien sind in Anhang A zu finden. In den nachfolgenden Abschnitten wird an den passenden Stellen erneut auf die einzelnen im Anhang abgedruckten Elemente des Lernmaterials verwiesen.

13 Umsetzung im Lernmaterial

Die Lesenden seien zudem eingeladen, die Lernplattform¹ zu besuchen, auf welcher das Lehr- und Lernmaterial in seiner vollen Funktion eingesehen werden kann.

¹Der Zugriff zur Lernplattform (workshops.cammp.online), die technischen Umsetzung sowie der grundsätzliche Aufbau der Materialien werden ausführlich in Kapitel 8 beschrieben.

Tabelle 13.1: Tabellarische Übersicht über das Lernmaterial zur Wortvorhersage
(Dauer in Minuten)

Phase	Inhalt	Dauer
Einstieg	Basiswissen zur Wortvorhersage und KI, Vereinfachung und erste Ideen zu Übersetzung der Realsituation in ein mathematisches Modell	15
Technik- Einführung		10
Arbeitsblatt 1	Erarbeitung und Erproben des Bi-Gramm-Modells	30
Besprechung 1	Sicherung von Arbeitsblatt 1, Sammeln der Probleme des Bi-Gramm-Modells und möglicher Verbesserungsvorschläge	15
Arbeitsblatt 2	Erarbeitung der Funktionsweise und Anwendung des Tri-Gramm-Modells, Anwendung des Uni-Gramm-Modells, Sammeln der Vor- und Nachteile der Modelle	20
Besprechung 2	Sicherung von Arbeitsblatt 2, Vergleich der Modelle und Sammeln von Ideen zur Kombination der Modelle	10
Arbeitsblatt 3	Kombination der Modelle: gewichteter Mittelwert der Übergangswahrscheinlichkeiten	30
Besprechung 3	Sicherung von Arbeitsblatt 3, Diskussion zur Wahl der Gewichte, Sammlung von Ideen zur Bewertung, Einführung der Kreuzentropie	20
Arbeitsblatt 4	Glättung des Sprachmodells, Bewertung von verschiedenen Sprachmodellen, Vergleich unterschiedlicher Modelle anhand der Kreuzentropie	70
Besprechung 4	Sicherung von Arbeitsblatt 4	10
Arbeitsblatt 5 <i>oder</i>	Formulierung eines Optimierungsproblems zur Bestimmung der optimalen Gewichte	20
Arbeitsblatt 5 <i>open</i>	Zusätzlich Entwickeln eines Optimierungsverfahrens	40
Besprechung 5 mit Abschluss	Sicherung von Arbeitsblatt 5, Vorstellung der Optimierungsverfahren (nur bei AB 5 open), Interpretation der Ergebnisse des Optimierungsverfahrens, Sammlung von Grenzen des Modells, Diskussion zu ethisch bedenklichen/ problematischen Aspekten bei der Nutzung und Entwicklung von Sprachmodellen	10
Zusatzblatt 3	Erarbeitung der Rückfallstrategie	15
Zusatzblatt 4	Untersuchung der Rolle der Qualität der Trainingsdaten	15
Zusatzblatt 5	Nutzung der Wortvorschläge zur Textgenerierung	30

13.1 Vorwissen und Lernziele

Im Lernmaterial entwickeln und evaluieren die Lernenden ein Sprachmodell und lernen die dafür notwendigen mathematischen Methoden kennen. Dabei sind teilweise mathematische Inhalte erforderlich, die über das Schulniveau hinausgehen. Diese werden im Material problembezogen erarbeitet und angewendet. Zugleich soll die mathematische Modellierung als Metakonzept für eine mathematische Herangehensweise an reale Probleme kennengelernt werden. Konkret sollen die folgenden **Lernziele** erreicht werden:

Die Lernenden können...

- ... die relative Häufigkeit von Wortsequenzen in Trainingstexten bestimmen. (vgl. Tabelle 6.1 L1, L2)
- ... relative Häufigkeiten eines stochastischen Prozesses sowohl im Übergangsgraphen als auch als Übergangstabelle darstellen. (vgl. Tabelle 6.1 L1, L2)
- ... eine Übergangstabelle lesen und deren Einträge im Sachkontext interpretieren. (vgl. Tabelle 6.1 L1)
- ... einen Übergangsgraphen lesen und dessen Einträge im Sachkontext interpretieren. (vgl. Tabelle 6.1 L1)
- ... die Auswahl eines Wortes nach bereits geschriebenen Wörtern als ein Zufallsexperiment beschreiben. (vgl. Tabelle 6.1 L3, L5)
- ... Texte als Trainingsdaten eines Sprachmodells identifizieren und deren Rolle bei der Entwicklung eines Sprachmodells erläutern. (vgl. Tabelle 6.1 L4)
- ... getroffene Annahmen benennen, darunter die Beschränkung der Wortvorgeschichte auf die $N - 1$ vorherigen Wörter sowie die Annahmen, dass Sprache als statisch angesehen werden kann und dass der Trainingsdatensatz den Sprachgebrauch ausreichend gut repräsentiert. (vgl. Tabelle 6.1 L3)
- ... Probleme des Bi-Gramm-Modells nennen und Überlegungen zur Verbesserung anstellen. (vgl. Tabelle 6.1 L7)
- ... die Funktionsweise von Uni-, Bi- und Tri-Gramm-Modell vergleichen und unterschiedliche Vor- und Nachteile erläutern. (vgl. Tabelle 6.1 L5)
- ... verschiedene Methoden (Rückfall-Strategie, gewichteter Mittelwert) zur Verknüpfung der einzelnen N -Gramm-Modelle erklären und deren Vorteil gegenüber einzelner N -Gramm-Modelle erläutern. (vgl. Tabelle 6.1 L5)
- ... die Testdaten explorieren und deren Rolle bei der Evaluation von Sprachmodellen erläutern. (vgl. Tabelle 6.1 L2, L8)
- ... verschiedene Modelle erproben und bewerten. (vgl. Tabelle 6.1 L9)
- ... die Kreuzentropie als Qualitätsmaß für einzelne Beispiele bestimmen sowie deren Ergebnisse hinsichtlich der Eignung des Sprachmodells interpretieren. (vgl. Tabelle 6.1 L10, L11)

- ... ein Optimierungsproblem sowie dessen Randbedingungen für die Optimierung der Interpolationsgewichte formulieren. (vgl. Tabelle 6.1 L6)
- ... Probleme und Grenzen des entwickelten Modells benennen. (vgl. Tabelle 6.1 L9)
- ... zentrale ethische Aspekte hinsichtlich Entwicklung und Einsatz von Systemen zur Wortvorhersage und im Allgemeinen von textbasierten Assistenzsystemen nennen und beurteilen. (vgl. Tabelle 6.1 L12)

Um eine erfolgreiche Bearbeitung des Lernmaterials zu ermöglichen, wird Vorwissen zu verschiedenen mathematischen Inhalten vorausgesetzt. Eine detaillierte Einordnung dieser Inhalte sowie die Zuordnung zu den entsprechenden Jahrgangsstufen erfolgt im Abschnitt zur curricularen Anbindung (vgl. Abschnitt 13.3). Besonders relevant sind dabei Kenntnisse zu

- relativer und absoluter Häufigkeit,
- Wahrscheinlichkeitsbegriff,
- Mittelwert,
- mehrstufige Zufallsexperimente, Anwenden der Pfadregel,
- natürlicher Logarithmus einer Zahl, Logarithmusgesetze.

13.2 Erläuterung des Lehr- und Lernmaterials

Im Folgenden wird der Aufbau des entwickelten Lernmaterials skizziert und didaktisch erläutert. Die anschließenden Abschnitte übernehmen die in Kapitel 11 bei der Darlegung des theoretischen Hintergrunds eingeführte Notation, um eine konsistente Darstellung zu gewährleisten. Im Lernmaterial weichen manche Notationen ab, da sie an schulübliche Konventionen angepasst oder für Lernende verständlicher gestaltet wurden.

13.2.1 Vorüberlegungen zur didaktischen Reduktion des *N*-Gramm-Modells

Um die Funktionsweise des *N*-Gramm-Modells zugänglich zu machen eignet sich besonders das Bi-Gramm-Modell als Einstieg, da es relativ einfach zu verstehen ist: Der nächste Zustand hängt nur vom vorherigen ab. Dadurch bleibt die Funktionsweise überschaubar, während bereits ein System entwickelt werden kann, das – anders als das Uni-Gramm-Modell – sinnvolle Vorschläge liefert. Die dabei gewonnenen Erkenntnisse sind dann leicht auf Modelle höherer Ordnung übertragbar.

Statt wie in Abschnitt 11.1 des theoretischen Hintergrunds direkt eine gesamte Wortsequenz zu analysieren, wird die Komplexität reduziert. Es wird zunächst lediglich

das Auftreten des nächsten Wortes v_i nach der Wortvorgeschichte v_j in einem vorliegenden Trainingsdatensatz betrachtet. In diesem Datensatz sei für jedes $k = 1, \dots, s$ das Bi-Gramm (v_j, v_k) genau $c(v_j, v_k)$ -mal enthalten. Die relative Häufigkeit des Bi-Grammes (v_j, v_i) unter allen Bi-Grammen im Trainingsdatensatz, die mit v_j beginnen, ergibt sich dann als

$$h(v_j|v_i) := \frac{c(v_j, v_i)}{c(v_j, v_1) + \dots + c(v_j, v_s)}. \quad (13.1)$$

Für die Vorhersage des nächsten Wortes sind nur diejenigen Wortsequenzen relevant, bei denen das erste Wort der vorgegebenen Wortvorgeschichte entspricht. Daher können die betrachteten Wortsequenzen direkt wie oben geschehen auf diejenigen mit erstem Wort v_j eingeschränkt und deren relativen Häufigkeiten für eine Vorhersage des nächsten Wortes verglichen werden.

Im Lernmaterial kann man anstelle der allgemeinen Notation für die relative Häufigkeit von Bi-Grammen zunächst mit konkreten Beispielen arbeiten. So kann beispielsweise zuerst die relative Häufigkeit der Wortsequenz bestehend aus den Wörtern „ich“ und „mag“ unter allen Bi-Grammen mit erstem Wort „ich“ bestimmt werden. Dadurch kann der Formalismus reduziert werden. Wie dies auf der konkreten Ebene im Lernmaterial umgesetzt wird, erläutert Abschnitt 13.2.2.

Ein Vorteil dieses Zugangs ist die klare Trennung der Ebene der realen Welt oder auch der Ebene der Daten von der Modellebene. Zuerst wird die Realsituation mit den tatsächlich beobachteten Auftretenshäufigkeiten lediglich beschrieben. Anschließend kann erläutert werden, weshalb das Auswählen des nächsten Wortes in einem Modell unter gewissen Annahmen als Zufallsexperiment aufgefasst werden kann. Die relativen Häufigkeiten im Trainingstext, der als Stichprobe des Sprachgebrauchs dient, sind dann Schätzwerte für die Wahrscheinlichkeiten dieses Zufallsexperiments. Innerhalb dieses Modells kann man dann von Übergangswahrscheinlichkeiten sprechen und die zugehörige Verteilung für Prognosen nutzen. Da jedoch aufgrund des hohen kognitiven Anspruchs und des nicht vorhandenen Vorwissens zur Punktschätzung nicht formal hergeleitet wird, dass die relativen Häufigkeiten tatsächlich Schätzwerte der Übergangswahrscheinlichkeiten sind, wird im Lernmaterial bewusst auf den Begriff „Schätzen“ verzichtet und stattdessen von einer „Näherung“ gesprochen.

Warum der Schluss von den relativen Häufigkeiten auf Übergangswahrscheinlichkeiten überhaupt erfolgen sollte, liegt neben der besseren Trennbarkeit von realer Situation (beobachtete relative Häufigkeiten) und Modell (theoretische Übergangswahrscheinlichkeiten) zudem an zwei Aspekten: Zum einen kann so an Vorwissen zu Konzepten, die aus dem Unterricht nur im Kontext von Wahrscheinlichkeiten bekannt sind (insbesondere für das in Abschnitt 13.2.5 beschriebene Arbeitsblatt 4, bei welchem mehrstufige Zufallsexperimente und Pfadregeln thematisiert werden), angeknüpft werden. Zum anderen wird bei der Bewertung die Wahrscheinlichkeit für das Auftreten des Trainingstextes im Sinne einer Unsicherheit für das Eintreten eines Ereignisses verwendet, wodurch eine reine Häufigkeitsinterpretation der Übergangswahrscheinlichkeit nicht mehr intuitiv scheint.

Es wäre grundsätzlich auch möglich, die in (13.1) definierten relativen Häufigkeiten als Wahrscheinlichkeiten eines Laplace-Experiments zu deuten. Dieser Ansatz ist allerdings aus verschiedenen Gründen problematisch: Zum einen müssten gleiche Zweier-Wortsequenzen des Trainingsdatensatzes künstlich indiziert werden, sodass sie unterscheidbar sind. Erst dann könnten diese indizierten Wortsequenzen als gleich wahrscheinliche Ausgänge eines Laplace-Experiments aufgefasst werden. Diese Vorgehensweise ist kompliziert und für Lernende daher schwer zugänglich. Zum anderen ist eine Verallgemeinerung der Aussagen, die für den vorliegenden Trainingsdatensatz getroffen werden, auf die Grundgesamtheit aller Texte dieses Genres nicht möglich. Da für die Vorhersage für das nächste Wort bei neu formulierten Texten eine solche Verallgemeinerung erforderlich ist, ist die Laplace-Interpretation ungeeignet.

13.2.2 Arbeitsblatt 1: Wortvorhersage mit dem Bi-Gramm-Modell

Auf dem ersten Arbeitsblatt (vgl. Anhang A.1) erfolgt die Einführung des Bi-Gramm-Modells über den in den Vorüberlegungen in Abschnitt 13.2.1 beschriebenen Zugang. Es werden zunächst erste Modellideen entwickelt und diese im Anschluss mathematisiert. Abschließend wird das Modell erprobt, und es werden die Grenzen seiner Anwendung erkundet.

Erste Modellideen

Um die Lernenden in ihrer Lebenswelt abzuholen, tippen sie zunächst eine Nachricht auf ihrem Smartphone und identifizieren die angebotenen Eingabehilfen (vgl. Kapitel 10). Anschließend werden die verschiedenen Eingabehilfen im Plenum gesammelt und diskutiert. Der Fokus richtet sich dann gezielt auf die Wortvorhersage, deren Funktionsweise im Rahmen des Modellierungstages untersucht werden soll. Dabei wird gemeinsam reflektiert, inwiefern diese Funktion nützlich ist. Es erfolgt eine Einordnung eines solchen Systems, das Vorschläge für das nächste Wort generiert, als KI-System. Dazu wird der Turing-Test (vgl. Abschnitt 3.2) als historischer Ansatz vorgestellt, um Systeme – insbesondere solche zur Sprachverarbeitung – als intelligent zu klassifizieren.

Um erste Ideen mit den Lernenden zu sammeln, wie die Vorhersage des nächsten Wortes funktionieren kann, wird zunächst ein kleines Beispiel betrachtet: Ein Nutzer hat in seiner Vergangenheit bereits die drei Sätze

„Ich mag Physik. Ich mag Sport. Ich liebe Mathe.“

eingetippt. Es wird der Fall betrachtet, dass der Nutzer einen neuen Satz mit dem Wort „ich“ beginnt. Die Lernenden sollen begründen, welche Wörter diesem Nutzer basierend – auf seinen Texten in der Vergangenheit – sinnvoll vorgeschlagen werden können. Da bei der Texteingabe häufig mehrere Wörter vorgeschlagen werden können, soll auch deren Reihenfolge festgelegt werden. Ziel ist es, dass Lernende hier mit den absoluten Häufigkeiten argumentieren. So tritt das Wort „mag“ zweimal nach dem Wort „ich“ auf und das Wort „liebe“ einmal. Im Unterrichtsgespräch kann die

absolute Häufigkeit in eine relative Häufigkeit überführt werden: In zwei bzw. einem aus drei Fällen tritt das Wort „mag“ bzw. „liebe“ nach dem Wort „ich“ auf. Auf der Grundlage des vorgegebenen Textes würde man daher zuerst das Wort „mag“ und als nächstes das Wort „liebe“ vorschlagen.

Die hierbei verwendete Methode kann dem Maschinellen Lernen zugeordnet werden und gleichzeitig der Begriff des Maschinellen Lernens am Beispiel der Wortvorhersage eingeführt werden. Dazu werden ML-Methoden auf einem sehr reduzierten Niveau von regelbasierten Verfahren abgegrenzt: Bei einem regelbasierten Verfahren würde für jedes vorangegangene Wort eine eigene Regel aufgestellt werden, welches Wort als nächstes vorgeschlagen werden soll. Das Maschinelle Lernen hingegen umfasst Methoden aus dem Gebiet der Künstlichen Intelligenz, die das Lernen aus Daten als Grundprinzip haben (vgl. Abschnitt 3.2). Im Fall der Wortvorhersage bedeutet dies, dass die Häufigkeit, mit der ein Wort nach bereits geschriebenen Wörtern auftritt, aus gegebenen Daten ermittelt wird. Die Information über diese Häufigkeit wird anschließend für die Vorhersage genutzt.

Ⓜ Einordnung in den Modellierungskreislauf

Eine kurze Einführung des Modellierungskreislaufs erfolgt (vgl. Anhang A.10). Im ersten Schritt wird das reale Problem als die Entwicklung eines Sprachmodells, das für die Erzeugung eines Wortvorschlags genutzt werden kann, identifiziert. Anschließend wird der bereits durchgeführte Schritt eingeordnet: Zur Vereinfachung wurde angenommen, dass ausschließlich das letzte Wort das nachfolgende Wort beeinflusst. Zudem wird für eine leichtere Umsetzung im Programmcode festgelegt, dass die Groß- und Kleinschreibung keine Rolle spielt. Da zunächst nur das letzte Wort für die Vorhersage berücksichtigt wird und somit die Funktionsweise des Bi-Gramm-Modells im Fokus steht, kann die Komplexität der Modellierung reduziert werden. Später wird diese Annahme schrittweise erweitert beziehungsweise abgewandelt, wodurch der Modellierungsprozess in mehrere Durchläufe unterteilt wird. Der Modellierungskreislauf dient während des gesamten Modellierungsworkshops als Orientierung: In den Plenumsphasen werden die auf den Arbeitsblättern erarbeiteten Teilschritte in den Kreislauf eingeordnet, sodass deren Bedeutung im Gesamtprozess herausgestellt wird. Wie von Schukajlow et al. (2011, S. 42 - 43) vorgeschlagen, wird der Modellierungskreislauf dabei als metakognitives visuelles Instrument genutzt, das die Reflexion des Modellierungsprozesses unterstützen und den Aufbau von Metawissen zum Modellieren fördern soll. Auch in diesem Kapitel erfolgt an mehreren Stellen eine Einordnung in den Modellierungskreislauf, die für eine bessere Sichtbarkeit mit dem Symbol des umkreisten Buchstabens „M“ versehen wird.

Mathematisierung

Auf dem ersten Arbeitsblatt sollen die im Plenumsgespräch gemachten Überlegungen mathematisiert werden. In der ersten Aufgabe sind alle Übergänge zwischen zwei Wörtern des im Plenum besprochenen Beispiels gesammelt in einem Übergangsgraphen dargestellt (vgl. Abbildung 13.7). In diesem Übergangsgraphen fehlen die relativen Häufigkeiten, mit der ein Wort auf ein anderes im vorliegenden Text folgt. Sie sollen von den Lernenden ergänzt werden.

Übergangsgraphen sind an allgemeinbildenden Gymnasien in Baden-Württemberg nicht Teil des Curriculums (Ministerium BW, 2016d) und auch in weiteren Bundesländern werden sie wenn, dann erst in der Oberstufe thematisiert (Ministerium BW, 2016d, S. 18). Übergangsgraphen sind daher für die meisten Lernende unbekannt. Diese Darstellungsart wird daher auf dem ersten Arbeitsblatt ausführlich erklärt. Zudem ist es wichtig, dass bei der späteren Besprechung darauf geachtet wird, dass der Übergangsgraph nicht mit einem Baumdiagramm verwechselt wird und es zu Fehlinterpretationen kommt. Der Übergangsgraph weist zwar eine ähnliche Struktur wie das Baumdiagramm auf und stellt Übergänge zwischen einzelnen Zuständen dar, jeder Zustand wird aber nur einmal dargestellt, um Redundanz zu vermeiden. Um trotzdem alle Übergänge darstellen zu können, kann der Übergangsgraph daher im Gegensatz zum Baumdiagramm auch Zyklen enthalten (das heißt, es ist möglich, von einem Zustand direkt oder über mehrere Übergänge wieder zum Ausgangszustand zu gelangen). Anders als das Baumdiagramm hat der Übergangsgraph zudem keinen klaren Anfangspunkt. Das passt auch zur vorliegenden Situation besser: Je nachdem, welches Wort bereits geschrieben ist, sind andere Übergänge relevant.

Die Darstellung im Übergangsgraphen wird bei größeren Trainingstexten schnell unübersichtlich. Daher sollen Lernende auch den Umgang mit der *Übergangstabelle* als eine zweite Darstellungsform der relativen Häufigkeiten üben. Diese Darstellungsform kommt zudem der Form näher, in welcher die relativen Häufigkeiten im System abgespeichert werden. Abbildung 13.1 zeigt die Übergangstabelle für die vorgegebenen drei Sätze. Die Zeilen der Tabelle stehen dabei für das bereits getippte Wort (Wort 1), die Wortvorgeschichte, und die Spalten für das Folgewort (Wort 2).

Durch die Bearbeitung dieser Aufgaben wird sowohl die graphische als auch tabellarische Darstellungsform einer Markov-Kette kennengelernt. Zudem wird geübt, Informationen aus beiden Darstellungsformen zu entnehmen, sie zu vernetzen und die Darstellungen zu erstellen. Die Aufgaben eignen sich somit neben dem inhaltlichen Kompetenzzuwachs auch für die Förderung eines zielgerichteten Umgangs mit verschiedenen Darstellungsformen im Sinne der prozessbezogenen Kompetenzen des Bildungsplans (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 15).

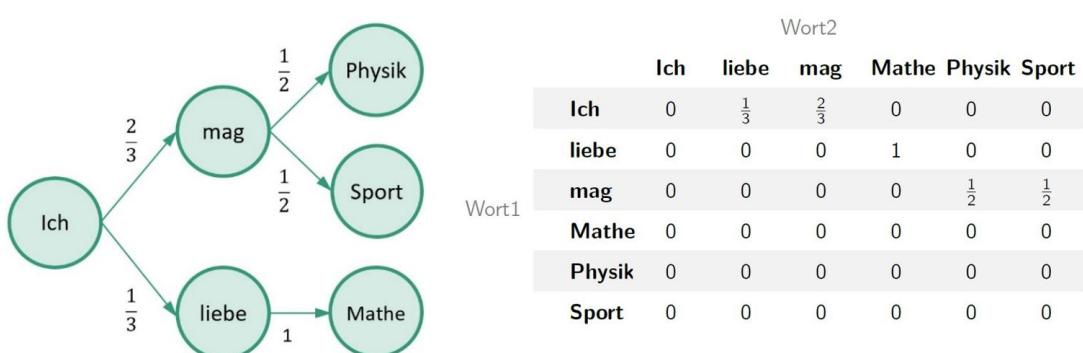


Abbildung 13.1: Übergangsgraph und Übergangstabelle des Bi-Gramm-Modells

Damit die relativen Häufigkeiten für größere Trainingsdatensätze nicht per Hand ermittelt werden müssen, sollen die Lernenden in der nächsten Aufgabe eine allgemeine Formel für die Bestimmung der relativen Häufigkeiten aufstellen. Um vom Beispiel leichter auf den allgemeinen Fall schließen zu können, werden sie zunächst aufgefordert, die Berechnung der relativen Häufigkeit $h_{\text{ich}}(\text{mag})$ des Übergangs von „ich“ zu „mag“ im vorangegangenen Beispiel formal zu notieren.

Insgesamt werden dazu alle Zweier-Wortsquenzen mit erstem Wort „ich“ betrachtet und deren jeweilige Auftretenshäufigkeit $c(\text{ich, ich})$, $c(\text{ich, liebe})$, $c(\text{ich, mag})$, $c(\text{ich, mathe})$, ... bestimmt. Die relative Häufigkeit ergibt sich dann zu

$$\begin{aligned} h(\text{mag}|\text{ich}) &= \frac{c(\text{ich, mag})}{c(\text{ich, ich}) + c(\text{ich, liebe}) + c(\text{ich, mag}) + \dots} \\ &= \frac{2}{0+1+2+0+0+0} = \frac{2}{3}. \end{aligned}$$

Im Allgemeinen lässt sich die relative Häufigkeit der aus Wort 1 und Wort 2 bestehenden Wortsequenz unter allen Wortsequenzen mit Wort 1 an erster Stelle somit durch

$$h(\text{Wort 2}|\text{Wort 1}) = \frac{c(\text{Wort 1, Wort 2})}{c(\text{Wort 1, Wort 1}) + c(\text{Wort 1, Wort 2}) + \dots}$$

bestimmen, was genau der in den Vorüberlegungen in Abschnitt 13.2.1 als sinnvoll erachteten Darstellung entspricht. Für eine leichtere Verständlichkeit wurden im Lernmaterial jedoch statt der Variablen v_i, v_j die Repräsentanten Wort 1 und Wort 2 verwendet.

Das eigentliche Ziel ist es allerdings, nicht nur den vorliegende Datensatz zu beschreiben, sondern die daraus ermittelte Information für Prognosen zu nutzen. Die relativen Häufigkeiten sollen also nicht nur zum Selbstzweck erhoben werden, sondern um Vorhersagen für das nächste Wort machen zu können. Dazu wird auf der Modell-ebene das Auswählen eines Wortes nach einem bereits geschriebenen Wort in einem festgelegten Genre von Texten als Zufallsexperiment aufgefasst. Die Trainingstexte sind dann Realisierungen dieses Zufallsexperiments. Da die Übergangswahrscheinlichkeit als Wahrscheinlichkeit für die Ausgänge dieses Zufallsexperiments unbekannt sind, wird die relative Häufigkeit als Näherung für die Übergangswahrscheinlichkeit verwendet. Es werden somit vom Trainingstext als Stichprobe Schlussfolgerungen auf die Grundgesamtheit – hier alle Texte eines Genres – gezogen. Dem liegt, wie bereits in Abschnitt 11.2 diskutiert, die Annahme zugrunde, dass die Trainingstexte die Gesamtheit aller Texte eines Genres – hier Texte die in Alltagssprache geschrieben werden – ausreichend gut repräsentieren. Zudem wird angenommen, dass diese Gesamtheit aller Texte und damit der Alltagswortschatz als statisch angesehen werden kann. Dieses Vorgehen wird Lernenden in einer Infobox erklärt und auch später nochmal in der Besprechungsphase aufgegriffen.

Dafür müssen allerdings geeignete Trainingsdatensätze verwendet werden. Für den aus den drei Beispielsätzen bestehenden Trainingsdatensatz ist die Annahme, dass

dieser repräsentativ sei, sicherlich nicht zutreffend. Deshalb ist es notwendig einen größeren Datensatz zu verwenden.

Insgesamt können damit in einem informellen problembezogenen Rahmen grundlegende Überlegungen der beurteilenden Statistik vermittelt werden: Für geeignete Prognosen sind Daten notwendig, die die beobachteten Ausgänge eines zufälligen Vorgangs darstellen. Für die Ausgänge dieses Vorgangs gibt es unbekannte theoretische Wahrscheinlichkeiten. Indem Muster in Daten identifiziert werden, versucht man, sich diesen unbekannten Wahrscheinlichkeiten zu nähern (Eichler & Vogel, 2009, S. 168).

Erproben des Modells

Das Bi-Gramm-Modell soll nun erprobt werden. Auf Basis der von den Lernenden erstellten Übergangstabelle werden die relativen Häufigkeiten automatisiert berechnet. Dieser Schritt ist dann das *Training* des Modells. Dasjenige Wort mit der größten relativen Häufigkeit wird schließlich ausgegeben. Erst wird das Modell mit dem aus obigen drei Sätzen bestehenden Trainingsdatensatz trainiert. Vorschläge können daher nur ausgegeben werden, wenn die Wortvorgeschichte im Trainingsdatensatz enthalten ist. Es wird daher bereits hier deutlich, dass bei geringen Datenmengen die Näherung der Übergangswahrscheinlichkeiten durch die relativen Häufigkeiten sehr schlecht sein kann.

Anschließend wird das Modell mit einem größeren Trainingsdatensatz trainiert. Auf Basis der entwickelten Formel werden die relativen Häufigkeiten automatisiert berechnet. Beim Ausprobieren dieses Modells haben die Lernenden die Aufgabe, Probleme zu identifizieren. Dazu werden folgende Reflexionsfragen gestellt:

- Erprobe das Modell mit einigen Wörtern. Beginne zunächst mit Wörtern, die im Sprachgebrauch häufiger vorkommen, und probiere dann auch Wörter aus, die seltener verwendet werden. Beschreibe, was du dabei beobachtest und erläutere, inwiefern es sich dabei um eine Schwäche des Modells handelt.
- Erprobe das Modell außerdem für eine Wortvorgeschichte, die länger als ein Wort ist. Prüfe vor allem, ob unterschiedliche Wörter vorgeschlagen werden, wenn das letzte Wort zwar gleich ist, die vorangegangenen Wörter aber unterschiedlich sind (z.B. „weißt du“ und „hast du“). Beschreibe deine Beobachtung. Erläutere, weshalb diese Beobachtung mit der Konstruktion des Modells erklärbar ist.

Anschließend erfolgt eine Plenumsphase, in der die getroffenen Annahmen für die Entwicklung eines Modells sowie die Berechnung der relativen Häufigkeit und die Erstellung des Übergangsgraphen besprochen werden. Nach dieser Sicherung können die Probleme des Modells, die die Lernenden bei der Erprobung beobachtet haben, gesammelt werden. Dazu zählen:

- (1) Für Wortvorgeschichten die nicht im Datensatz auftreten, kann kein nächstes Wort vorgeschlagen werden.
- (2) Die Wortvorschläge hängen nur vom vorherigen Wort ab und nicht vom erweiterten Satzkontext.

Daraufhin werden mögliche Modellverbesserungen mit den Lernenden gesammelt und diskutiert. Verbesserungen, die im Lernmaterial erarbeitet werden, sind zum einen die Erweiterung der Wortvorgeschichte um ein Wort (Tri-Gramm-Modell). Zum anderen kann das häufigste Wort im Datensatz ausgegeben werden, um bei Wortvorgeschichten, die nicht im Trainingstext aufgetreten sind, trotzdem einen Vorschlag machen zu können. Eine weitere naheliegende mögliche Verbesserung wäre die Verwendung eines größeren Trainingsdatensatzes.

13.2.3 Arbeitsblatt 2: Uni- und Tri-Gramm-Modell

⑮ Einordnung in den Modellierungskreislauf

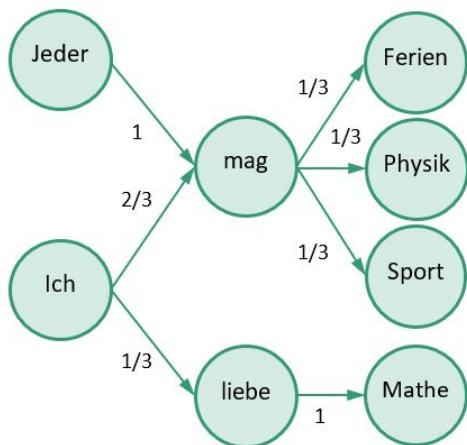
Das Uni- und Tri-Gramm-Modell werden auf Arbeitsblatt 2 (vgl. Anhang A.2) als alternative Modelle kennengelernt. Der Modellierungskreislauf wird somit zwei weitere Male durchlaufen, wobei jedoch lediglich die Annahme, dass nur das letzte Wort die Vorhersage beeinflusst, entsprechend angepasst wird.

Erweiterung des Trainingsdatensatzes

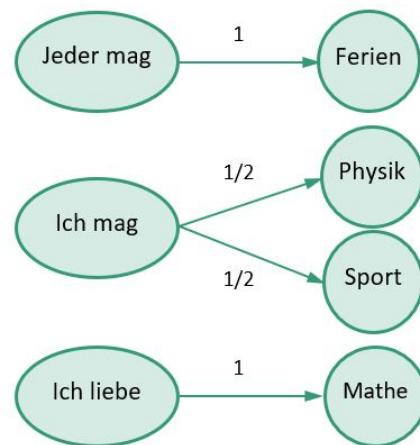
Um die Vorteile des Tri-Gramm-Modells überhaupt erkennen zu können, wird der Trainingsdatensatz aus dem Beispiel im ersten Arbeitsblatt um einen zusätzlichen Satz erweitert:

„Ich mag Physik. Ich mag Sport. Ich liebe Mathe. Jeder mag Ferien.“

Zunächst soll der Übergangsgraph des Bi-Gramm-Modells zum erweiterten Datensatz erstellt werden (vgl. Abbildung 13.2(a)), sodass später Vergleiche zwischen dem



(a) im Bi-Gramm-Modell



(b) im Tri-Gramm-Modell

Abbildung 13.2: Übergangsgraphen zum aus vier Sätzen bestehenden Trainingsdatensatz in verschiedenen Modellen

Bi-Gramm-Modell und dem Tri-Gramm-Modell auf Basis des gleichen Datensatzes möglich sind. Für die Erstellung des Übergangsgraphen des Bi-Gramm-Modells zum erweiterten Trainingsdatensatz sind nur geringfügige Anpassungen am bisherigen Übergangsgraphen erforderlich: Einerseits müssen die neuen Zustände hinzugefügt, andererseits die relativen Häufigkeiten neu berechnet werden. Dadurch nimmt die Bearbeitung der Aufgabe nicht viel Zeit in Anspruch, trägt aber dazu bei, das auf Arbeitsblatt 1 Gelernte zu festigen. Im Übergangsgraphen soll anschließend abgelesen werden, welche Vorschläge nach der Wortvorgeschichte „Jeder mag“ beziehungsweise „Ich mag“ auf Basis dieses Modells gegeben werden.

Tri-Gramm-Modell

Anschließend soll der Übergangsgraph des Tri-Gramm-Modells erstellt werden (vgl. Abbildung 13.2(b)). Da die Wortvorgeschichte in diesem Modell aus zwei Wörtern besteht, werden jeweils zwei aufeinanderfolgende Wörter als ein Knoten im Graphen dargestellt. Eine Ausnahme bilden die Knoten in Abbildung 13.2(b), die das letzte Wort eines Satzes repräsentieren: Hier wird nur das letzte Wort erfasst, da der Satz danach endet und der Knoten somit keine Wortvorgeschichte für ein weiteres Wort mehr bildet. Ein Vorteil dieser Darstellung ist, dass alle rechten Knoten stets ein Folgewort zur Wortvorgeschichte im linken Knoten darstellen, sodass mögliche Vorschläge für das nächste Wort leicht ablesbar sind. Alternativ wäre es jedoch auch korrekt, die letzten beiden Wörter eines Satzes gemeinsam in einem Knoten darzustellen – im ersten Satz beispielsweise „mag Ferien“. Zudem werden die relativen Häufigkeiten berechnet und im Übergangsgraphen ergänzt.

Anschließend soll auch das Tri-Gramm-Modell erprobt werden. Zunächst soll im Übergangsgraphen abgelesen werden, welche Vorschläge nach der Wortvorgeschichte „Jeder mag“ beziehungsweise „Ich mag“ auf Basis des Tri-Gramm-Modells gegeben werden. Die gleiche Aufgabe hatten die Lernenden bei der Erprobung des Bi-Gramm-Modells, sodass es möglich ist, die Ergebnisse direkt miteinander zu vergleichen. Weitere Erprobungen des mit einem größeren Trainingsdatensatz trainierten Tri-Gramm-Modells erfolgen. Dabei soll insbesondere überprüft werden, ob unterschiedliche Wörter vorgeschlagen werden, wenn das letzte Wort der Wortvorgeschichte zwar gleich ist, die vorletzten Wörter aber unterschiedlich sind. Es sollen aber auch alle anderen Beobachtungen notiert werden, zum Beispiel, wie häufig kein Vorschlag gemacht werden kann.

Uni-Gramm-Modell

Da die Modellierung im Uni-Gramm-Modell keine neuen mathematischen Erkenntnisse beisteuert, wird dieses Modell lediglich von den Lernenden erprobt. Auch hier sollen die Beobachtungen mit der Funktionsweise des Modells erklärt werden.

Vergleich der Modelle

Abschließend sollen die Modelle verglichen und Vor- und Nachteile eines jeden Modells identifiziert werden, wobei gleichzeitig das Verständnis der Konzepte vertieft wird. Die Vor- und Nachteile der einzelnen Modelle werden im Plenum gesammelt und besprochen (vgl. Abbildung 13.3). Zusammenfassend wird festgehalten:

- Je länger die Wortvorgeschichte, desto besser passt der Wortvorschlag zum Satzkontext.

Modell	Vorteile	Nachteile
Uni-Gramm-Modell	immer ein Vorschlag, wenig Rechenaufwand	Vorschläge nicht kontextbasiert
Bi-Gramm-Modell	Vorschläge beziehen den Kontext nur wenig mit ein	geringer Kontextbezug, manchmal kein Vorschlag durch ungesehene Wortvorgeschichten
Tri-Gramm-Modell	Vorschläge beziehen den Kontext mit ein	häufig kein Vorschlag durch ungesehene Wortvorgeschichten, hoher Rechenaufwand

Abbildung 13.3: Bildschirmfoto der Präsentationsfolie zum Vergleich der N -Gramm-Modelle

- Je kürzer die Wortvorgeschichte, desto weniger ungesehene Wortvorgeschichten gibt es, und desto verlässlicher ist die Näherung der Übergangswahrscheinlichkeiten mit den relativen Häufigkeiten aufgrund höherer Auftretenshäufigkeiten der Wortsequenzen.

Bearbeiten die Lernenden die Zusatzaufgabe, bei der ermittelt werden soll, wie viele Uni-, Bi- und Tri-Gramme ohne Berücksichtigung grammatischer Regeln möglich sind (vgl. Abschnitt 11.3), kann dieses Wissen gezielt für den Modellvergleich genutzt werden – insbesondere für eine stärker quantitative Abschätzung der Anzahl ungesehener Wortvorgeschichten. Für eine verlässliche und zugleich kontextbasierte Wortvorhersage ist deshalb eine Kombination der Modelle notwendig.

13.2.4 Arbeitsblatt 3: Kombination der N -Gramm-Modelle

⑩ Einordnung in den Modellierungskreislauf

Bei der Diskussion von Vor- und Nachteile der drei N -Gramm-Modelle sollen die Lernenden erkennen, dass alle Modelle unterschiedliche Stärken und Schwächen haben und für eine verlässliche und zugleich kontextbasierte Vorhersage des nächsten Wortes die N -Gramm-Modelle kombiniert werden müssen. Die Kombination der Modelle kann dann auf Arbeitsblatt 3 (vgl. Anhang A.4) als Modellverbesserung eingeführt werden.

Im Plenum werden zunächst Ideen zur Kombination der Modelle gesammelt (vgl. Abschnitt 14.1). Als in der Praxis gängige Verfahren werden die Rückfallstrategie und die Interpolation der einzelnen Modelle benannt. Die Interpolation der Modelle wird anschließend in Arbeitsblatt 3 erarbeitet, während die Rückfallstrategie auf einem Zusatzblatt entwickelt werden kann (vgl. Abschnitt 13.2.7).

Im Lernmaterial wird die Interpolation der N -Gramm-Modelle als Kombination über einen gewichteten Mittelwert bezeichnet. Der Begriff der linearen Interpolation ist

zwar in der Sprachverarbeitung gängig (Chen & Goodman, 1998, S. 10; Fink, 2003, S. 109, Wendemuth, 2004, S. 31), wird im Lernmaterial aber nicht verwendet. Zum einen dürfte dieser unbekannt sein, und zum anderen steht bei der Kombination der Verteilungen der einzelnen Modelle nicht die Vorstellung als Approximationsproblem zu vorgegebenen Stützpunkten im Vordergrund. Stattdessen wird die Kombination der Modelle am Beispiel einer *Gesamtübergangswahrscheinlichkeit* eines bestimmten Übergangs eingeführt. Die Gesamtübergangswahrscheinlichkeit ergibt sich dann aus dem gewichteten Mittelwert der einzelnen Übergangswahrscheinlichkeiten.

Der Begriff des gewichteten Mittelwerts ist Lernenden aus dem Schulunterricht in der Regel ebenfalls nicht bekannt, auch wenn er implizit in verschiedenen Kontexten (Erwartungswertbildung, Berechnung der Gesamtnote eines Kurses) verwendet wird. An dieses Vorwissen wird im Lernmaterial angeknüpft. Die Gesamtnote eines Kurses wird als Beispiel für die Bildung eines gewichteten Mittelwerts aufgegriffen, um den gewichteten Mittelwert auf Arbeitsblatt 3 einzuführen². Dieses Beispiel eignet sich besonders gut, da es aus dem Alltag vertraut ist und somit einen anschaulichen und leicht verständlichen Zugang zum Konzept des gewichteten Mittelwerts bietet.

Konkret soll die Gesamtnote eines Kurses berechnet werden, in welchem Klassenarbeiten und Tests geschrieben wurden, die wie die mündliche Note einen jeweils unterschiedlichen Anteil der Gesamtnote ausmachen. Dabei werden konkrete Zahlenwerte für die Anteile und die Noten vorgegeben. Von diesem Beispiel sollen Lernende schließlich auf die allgemeine Formel des gewichteten Mittelwertes

$$g_1 \cdot n_1 + g_2 \cdot n_2 + g_3 \cdot n_3$$

mit den Werten n_1, n_2, n_3 und ihren jeweiligen Gewichten g_1, g_2, g_3 mit $0 \leq g_1, g_2, g_3 \leq 1$ und $g_1 + g_2 + g_3 = 1$ schließen. Dieses Wissen kann anschließend genutzt werden, um eine Formel für die Gesamtübergangswahrscheinlichkeit des Folgewortes w_3 mit der Wortvorgeschichte w_1, w_2 aufzustellen³:

$$p*_{int}(w_3|w_1, w_2) = g_1 \cdot p(w_3) + g_2 \cdot p(w_3|w_1) + g_3 \cdot p(w_3|w_1, w_2).$$

Da die Berücksichtigung ungesehener Wörter für die Nutzung des Sprachmodells zur Vorhersage nicht relevant ist, wird sie hier noch nicht vorgenommen. Stattdessen wird die Glättung auf Arbeitsblatt 4 (vgl. Abschnitt 13.2.5) bei der Bewertung des Sprachmodells thematisiert und hier zunächst nur das Uni-, Bi- und Tri-Gramm-Modell interpoliert.

Anschließend sollen erste Überlegungen zur Wahl der Gewichte angestellt werden und dazu, welche Werte für die Gewichte ausgeschlossen werden können. Neben der Randbedingung, dass die Gewichte Werte von null bis eins annehmen können, die bereits bei der allgemeinen Formel des gewichteten Mittelwertes erarbeitet wurde, kön-

²Es ist natürlich klar, dass die Notenbildung keine rein mathematische Berechnung, sondern eine pädagogisch fachliche Gesamtwertung ist und die Durchschnittsnote lediglich eine Orientierung gibt. Um hier kein falsches Bild zu vermitteln, wird auf diese Tatsache in der Aufgabenstellung hingewiesen.

³Die Fallunterscheidung für $i = 1$ und $i = 2$ (vgl. Abschnitt 11.3.2) wird aus Gründen der Übersichtlichkeit vernachlässigt und nur im Hintergrundcode berücksichtigt.

nen weitere heuristische Überlegungen gemacht werden. Zum einen sollten die Werte null und eins ausgeschlossen werden. Andernfalls würden nicht alle Modelle berücksichtigt, was den Vorüberlegungen zur Kombination der Modelle widerspricht. Zum anderen – und diese Überlegung kann später ausführlich im Plenum mit den Lernenden besprochen werden – können bereits Überlegungen angestellt werden, welche Übergangswahrscheinlichkeit am stärksten und welche am wenigsten stark gewichtet werden sollte. Beim Vergleich der einzelnen Modelle zeigte sich, dass die Vorschläge des Tri-Gramm-Modells, gefolgt von denen des Bi-Gramm-Modells, am präzisesten und stärksten kontextbezogen sind, sofern die Wortvorgeschichte im Trainingsdatensatz enthalten ist. Dank der Integration des Uni-Gramm-Modells im kombinierten Ansatz stellt es nun kein Problem mehr dar, wenn eine Wortvorgeschichte nicht im Trainingsdatensatz enthalten ist. In solchen Fällen kann ein Vorschlag auf Basis der Übergangswahrscheinlichkeiten des Uni-Gramm-Modells generiert werden. Es erscheint daher intuitiv sinnvoll, die Übergangswahrscheinlichkeit des Tri-Gramm-Modells stärker als die des Bi-Gramm-Modells und die des Uni-Gramm-Modells zu gewichten. Welche Gewichtung sich jedoch konkret als sinnvoll erweist, hängt stark davon ab, wie oft die Uni-, Bi- und Tri-Gramme im Trainingsdatensatz beobachtet werden und auf welchen absoluten Auftretenshäufigkeiten die relativen Häufigkeiten somit basieren. Die Gewichtung ist folglich stark vom vorliegenden Trainingsdatensatz beeinflusst. Genauere Aussagen sind deshalb an dieser Stelle nicht möglich.

Im Anschluss wird beispielhaft für einen Übergang ermittelt, welche Gesamtübergangswahrscheinlichkeit das Modell diesem Übergang zuweist. Zudem haben die Lernenden die Möglichkeit, sich für eine vorgegebene Wortvorgeschichte Vorschläge ausgeben zu lassen. Dabei sollen verschiedene Gewichtungen ausprobiert werden.

In der anschließenden Besprechungsphase werden die gewonnenen Erkenntnisse zur Wahl der Gewichte reflektiert. Dabei wird festgehalten, dass bislang lediglich sinnvolle Wertebereiche und eine plausible Rangfolge für die Gewichte identifiziert wurden. Da genauere Aussagen nicht möglich sind, stellt sich zwangsläufig die Frage, nach welchen Kriterien die Gewichte bestimmt werden sollten. Ziel ist es, die Gewichte so zu wählen, dass das Sprachmodell möglichst „gut“ ist. Dies führt unmittelbar zur zentralen Frage: „Wann bezeichnet man ein Sprachmodell als „gut“?“

13.2.5 Arbeitsblatt 4: Bewertung des Sprachmodells

Ⓜ Einordnung in den Modellierungskreislauf

Die Frage nach den optimalen Interpolationsgewichten motiviert die Definition eines geeigneten Qualitätsmaßes. Im Plenumsgespräch und auf Arbeitsblatt 4 (vgl. Anhang A.4) wird die Kreuzentropie als Qualitätsmaß erarbeitet und dabei die Rolle der Testdaten bei der Evaluation von Sprachmodellen erläutert. Anschließend wird die Kreuzentropie für interpolierte N -Gramm-Modelle mit verschiedenen Gewichtungen ermittelt, und die Sprachmodelle werden auf Basis dieses Qualitätsmaßes verglichen.

Ideen zur Bewertung

Im Unterrichtsgespräch werden zunächst die Ideen der Lernenden zur Bewertung

von Sprachmodellen gesammelt. Zu den möglichen Kriterien für die Bewertung eines Sprachmodells, das zur Wortvorhersage genutzt wird, gehören unter anderem:

- die Zeitersparnis durch die Nutzung der Wortvorschläge,
- die Einsparung von Tastenanschlägen durch die Nutzung der Wortvorschläge,
- die relative Häufigkeit, mit der ein Wort tatsächlich ausgewählt wird (Trefferrate),
- die Kreuzentropie.

Diese hier dargestellten Kriterien orientieren sich an in der Forschung verwendeten Qualitätsmaßen (Carstensen et al., 2009, S. 127; Väyrynen et al., 2007, S. 481; Wandmacher, 2008, S. 28), wobei darauf geachtet wurde, diejenigen aufzunehmen, die besonders häufig von Lernenden benannt wurden (vgl. Abschnitt 14.1).

Als Nächstes lässt sich die Eignung dieser Kriterien als Qualitätsmaß diskutieren. Die ersten beiden Qualitätsmaße eignen sich nur, um das Sprachmodell für diesen speziellen Anwendungszweck zu bewerten, und sie sind aufwändiger zu ermitteln. Insbesondere sind für die Messung der Zeitersparnis tatsächlich Testpersonen notwendig. Da die Messung der Zeitersparnis tatsächlich den Einsatz von Testpersonen erfordert, bietet sich dies als Ausgangspunkt an, um zu erklären, dass die Bewertung mit Testdaten oft einfacher ist. Daraufhin wird die Rolle der Testdaten im ML-Prozess erläutert.

Testdaten werden beispielsweise zur Bestimmung der Trefferrate oder der Kreuzentropie genutzt. Um die Trefferrate zu ermitteln, wird fortlaufend ein Wortvorschlag generiert, während der Testdatensatz – ein einfacher Text – schrittweise aufgedeckt und das aufgedeckte Wort mit dem Wortvorschlag verglichen wird. In diesem Zusammenhang ist es entscheidend, zwischen Trainingsdaten und Testdaten zu unterscheiden: Die Trainingsdaten werden zur Berechnung der relativen Häufigkeiten genutzt. Die Testdaten dienen zur Bewertung des Sprachmodells. Sie sind im Vergleich zu den Trainingsdaten unbekannt – das heißt nicht zur Erzeugung des Modells verwendete – Daten, um den Anwendungsfall simulieren zu können. Die Rolle der Test-

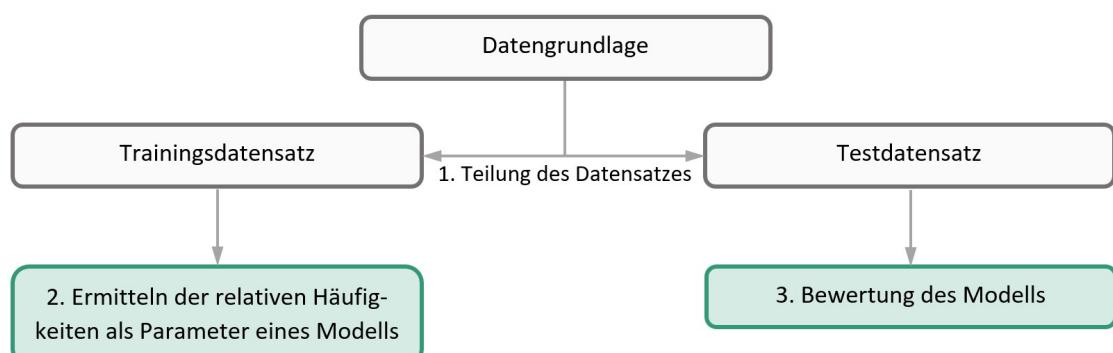


Abbildung 13.4: Rolle der Test- und Trainingsdaten im N-Gramm-Modell

und Trainingsdaten wird den Lernenden anhand des Schaubildes in Abbildung 13.4 erläutert.

Kreuzentropie als Qualitätsmaß

Auch für die Kreuzentropie wird erklärt, wie die Testdaten hier genutzt werden. Hierzu erfolgt zunächst eine Vorstellung der Kreuzentropie. Sie kann, ohne für Lernende durchaus komplexe Konzepte aus der Informationstheorie verwenden zu müssen, über die Likelihoodfunktion $P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_n = w_n)$ eingeführt werden, welche im Lernmaterial nach dem Vorbild von Carstensen et al. (2009, S. 130) kurz mit *Modellwahrscheinlichkeit* bezeichnet wird. Im Unterrichtsgespräch wird die Funktion der einzelnen Bestandteile der Kreuzentropie

$$K_T(P_{g_0, g_1, g_2, g_3}) = -\frac{1}{n} \log P_{g_0, g_1, g_2, g_3}(X_1 = w_1, \dots, X_n = w_n)$$

erarbeitet:

- Die Modellwahrscheinlichkeit gibt an, wie wahrscheinlich ein Testdatensatz bestehend aus den Wörtern w_0 bis w_n aus einem zugrunde liegenden Sprachmodell erzeugt wird. Das Erzeugen des Testdatensatzes mit einem Sprachmodell wird als mehrstufiges Zufallsexperiment aufgefasst. Nach jeder Wortsequenz hat das darauffolgende Wort eine durch das Sprachmodell vorgegebene Wahrscheinlichkeit, mit der es auftritt. Das kann anhand eines konkreten Testdatensatzes veranschaulicht werden (vgl. Abbildung 13.5). Auf dem Arbeitsblatt haben die Lernenden die Aufgabe, die Modellwahrscheinlichkeit als Ausdruck der einzelnen Übergangswahrscheinlichkeiten zu formulieren (erste Pfadregel). Die zugehörige Aufgabe wird später im Detail erläutert. Auf die Indexierung der Wahrscheinlichkeit mit den Parametern des Modells wird für eine einfachere Darstellung im Lernmaterial verzichtet und der Umstand stattdessen im Text beschrieben.
- Die Skalierung mit der Größe des Testdatensatzes n ist notwendig, damit man sich keinen Vorteil verschaffen kann, wenn man nur einen sehr kleinen Testdatensatz mit zum Beispiel nur drei oder vier Wörtern auswählt.
- Das negative Vorzeichen kann motiviert werden, indem gezeigt wird, dass der Wert der Kreuzentropie ansonsten negativ wäre. Nimmt das Qualitätsmaß positive Werte an, so kann das jedoch die Interpretation erleichtern. (Andernfalls wäre unklar, ob mit „kleineren Werten“ gute Modelle durch weniger negative oder durch stärker negative Zahlen ausgedrückt werden – was zu Verwirrung führen könnte.) Um zu veranschaulichen, weshalb die Kreuzentropie ohne das Vorzeichen stets negativ wäre, wird der Graph der Logarithmusfunktion betrachtet. Die Lernenden werden dann aufgefordert, den Definitionsbereich der Funktion, der hier relevant ist, zu benennen sowie die Werte, die die Funktion für diesen Bereich annimmt.
- Die Notwendigkeit des Logarithmierens der Modellwahrscheinlichkeit wird im zugehörigen Arbeitsblatt erarbeitet. Eine mögliche Herleitung über die Kreuzentropie aus der Informationstheorie (vgl. Abschnitt 11.4) wird bewusst nicht gewählt, da sie die Einführung zusätzlicher Konzepte erfordern würde, die über

den vorgesehenen Rahmen hinausgehen. Stattdessen wird das Logarithmieren im Kontext der numerischen Stabilität motiviert und von den Lernenden als Maßnahme zur Vermeidung eines numerischen Unterlaufs identifiziert. Bei einem numerischen Unterlauf handelt es sich um einen Rundungsfehler, der passt, wenn Zahlen nahe null auf null abgerundet werden (Goodfellow et al., 2018, S. 87 - 88). Bereits im Unterrichtsgespräch erhalten die Schülerinnen und Schüler jedoch den Hinweis, dass die Modellwahrscheinlichkeit sehr klein werden kann, was für deren Berechnung mit dem Computer ein Problem ist, da die Zahl dann nicht mehr darstellbar ist. Am Graph der Logarithmusfunktion kann überlegt werden, welchen Wert der Logarithmus für eine sehr kleine Zahl hat. Wie viel hier bereits gemeinsam erarbeitet wird und wie viel man die Lernenden sich anschließend im Arbeitsblatt selbst erschließen lässt, sollte abhängig von der Leistungsstärke des Kurses gemacht werden. Neben der Vermeidung numerischer Probleme hat das Logarithmieren zudem den Vorteil, dass es die mathematische Darstellung vereinfacht, insbesondere wenn die Kreuzentropie als Fehlerfunktion genutzt wird und ein Minimum dieser zum Beispiel über das Differenzieren der Fehlerfunktion bestimmt werden soll. Da die Kreuzentropie hier zunächst nur als Qualitätsmaß eingesetzt wird, wird dieser Vorteil der logarithmischen Transformation nicht explizit ersichtlich und daher auch nicht im Workshop thematisiert.

Die Kreuzentropie hat gegenüber der Trefferrate den Vorteil, dass deren Berechnung weniger aufwändig ist. Bei der Trefferrate müsste für jede folgende Wortvorgeschichte im Testdatensatz ein Vorschlag über das Sprachmodell ermittelt werden und somit ein Vergleich der Übergangswahrscheinlichkeiten aller möglichen Übergänge stattfinden, während bei der Kreuzentropie immer nur die Übergangswahrscheinlichkeit für den konkreten im Testdatensatz auftretenden Übergang ermittelt wird. Zusätzlich eignet sich die Kreuzentropie nicht nur als Qualitätsmaß für die Bewertung von Sprachmodellen, die zur Wortvorhersage genutzt werden, sondern auch für die Bewertung von Sprachmodellen unabhängig von deren Anwendungsbereich.

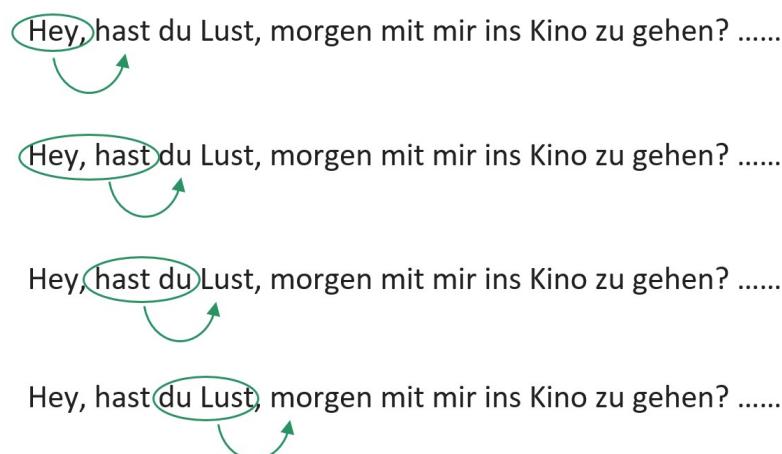


Abbildung 13.5: Veranschaulichung der Ermittlung der Modellwahrscheinlichkeit an einem konkreten Testdatensatz

Aus mathematischer Sicht hat die Kreuzentropie zudem den Vorteil, dass sie eine stetige, konvexe Funktion der Parameter des Sprachmodells darstellt (vgl. Abschnitt 11.5) und sich somit auch als Fehlerfunktion eignet. Dieser Aspekt wird mit den Lernenden allerdings nicht besprochen. Es kann allerdings später, wenn die Kreuzentropie als Fehlerfunktion eingesetzt wird, die Information gegeben werden, dass die Kreuzentropie auch deshalb eine gute Wahl ist, weil sie als Funktion der Gewichte formuliert werden kann und damit eine Möglichkeit bietet, die Gewichte sinnvoll festzulegen.

Auf dem vierten Arbeitsblatt soll die Bewertung des entwickelten Sprachmodells mithilfe der Kreuzentropie erfolgen. Unter Nutzung der ersten Pfadregel (oder auch Produktregel) stellen die Lernenden zunächst die Modellwahrscheinlichkeit unter dem interpolierten Tri-Gramm-Modell als Produkt der einzelnen Übergangswahrscheinlichkeiten dar:

$$P_{0,g_1,g_2,g_3}(X_1 = w_1, \dots, X_N = w_n) = \prod_{i=1}^n p_{int}(w_i | w_{i-2}, w_{i-1}).$$

Im Lernmaterial wird hier, die an die Wahrscheinlichkeit $p_{int}(w_i | w_{i-2}, w_{i-1})$ geknüpfte Bedingung nicht explizit notiert, sondern lediglich im begleitenden Text dargelegt. Anstelle des Produktzeichens wird die Multiplikation durch eine explizite Darstellung einiger zentraler Faktoren – insbesondere des ersten und letzten – verdeutlicht, während die dazwischenliegenden Faktoren durch drei Punkte angedeutet werden (vgl. Anhang A.4). Durch diese reduzierte Notation wird die Darstellung erheblich vereinfacht, sodass der mathematische Kern klarer ersichtlich ist und nicht durch die formale Schreibweise verdeckt wird.

Um die Berechnung der Wahrscheinlichkeit im mehrstufigen Zufallsexperiment zu vereinfachen, kann das Zeichnen der ersten Stufen des zugehörigen Baumdiagramms hilfreich sein. Um das Baumdiagramm übersichtlich darstellen zu können, werden nicht relevante Pfade weggelassen (vgl. Abbildung 13.6). Diese Vorgehensweise empfehlen auch Krüger, Sill und Sikora (2015, S. 142 – 143), um komplexe mehrstufige Zufallsexperimente im Unterricht anschaulicher darzustellen. Brauchen Lernende bei der Aufgabe Unterstützung, so wird ihnen dieser Ansatz durch gestufte Tipps vermittelt. Alternativ kann das Erzeugen einer Wortsequenz, wie auf Arbeitsblatt 1 und 2 geschehen, genauso gut als stochastischer Prozess betrachtet werden, bei welchem die Ausgänge aggregiert werden. Allerdings ist das Zeichnen eines Übergangsgraphen an dieser Stelle nicht mehr praktikabel, und die Gesetzmäßigkeiten stochastischer Prozesse sind den Lernenden in allgemeinbildenden Gymnasien nicht aus dem Schulunterricht bekannt. Daher wurde der Zugang über mehrstufige Zufallsexperimente gewählt, um an das Vorwissen der Lernenden anknüpfen zu können. Je nach Vorbildung der Lerngruppe können und sollten hier Anpassungen vorgenommen werden. Insbesondere sollte verdeutlicht werden, dass die in den Tipps vorgeschlagene Verwendung eines Baumdiagramms nur ein möglicher Lösungsweg ist und alternative Zugänge ebenfalls denkbar sind.

Anschließend kann der Frage nachgegangen werden, weshalb die Modellwahrscheinlichkeit für einen Testdatensatz mit mehreren Wörtern einen sehr kleinen Wert hat:

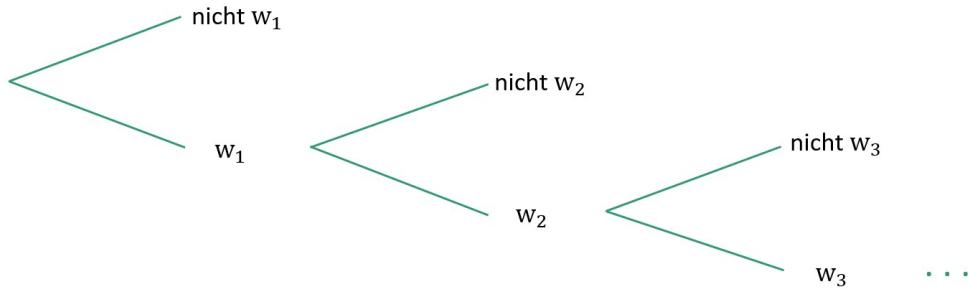


Abbildung 13.6: Baumdiagramm zur Ermittlung der Modellwahrscheinlichkeit

Die Modellwahrscheinlichkeit ergibt sich aus dem Produkt vieler Übergangswahrscheinlichkeiten, die zwischen null und eins liegen. Da jede Multiplikation mit einem Wert kleiner als eins das Ergebnis weiter verkleinert, wird die Modellwahrscheinlichkeit oft extrem klein, und es kommt zu einem numerischen Unterlauf. Um dieses Problem zu vermeiden, wird die Modellwahrscheinlichkeit logarithmiert, sodass das Produkt als eine Summe dargestellt werden kann. Die Lernenden haben dann die Aufgabe, unter Nutzung der Logarithmusgesetze die Umformung

$$\log \prod_{i=1}^n p_{int}(w_i | w_{i-2}, w_{i-1}) = \sum_{i=1}^n \log p_{int}(w_i | w_{i-2}, w_{i-1})$$

vorzunehmen. Diese Summe ist für den Computer leichter zu berechnen, und Rundungsfehler werden reduziert.

Konkrete Berechnung der Kreuzentropie und Einführung der Glättung
 Bevor die Kreuzentropie für das interpolierte Tri-Gramm-Modell relativ zu einem konkreten Testdatensatz ermittelt wird, erfolgt zunächst eine Erkundung des Testdatensatzes und dessen Struktur. Die Lernenden haben die Aufgabe, Teile des Testdatensatzes ausgeben zu lassen, explorieren somit die Datengrundlage und lernen gleichzeitig Befehle für den Zugriff auf einzelne Elemente kennen, die sie im Anschluss für die Berechnung der Modellwahrscheinlichkeit verwenden. Mithilfe einer for-Schleife, in der sie Start- und Endwert sowie einen Platzhalter im Schleifenrumpf ersetzen müssen, wird die logarithmierte Modellwahrscheinlichkeit unter dem interpolierten Tri-Gramm-Modell berechnet (vgl. Anhang A.1). Zur Unterstützung für diejenigen, die keine Programmierkenntnisse haben, stehen ausklappbare Tipps sowie eine ausführliche Tipp-Karte zur Struktur und Funktionsweise der for-Schleife zur Verfügung.

Da das Modell bislang nicht geglättet wurde, stoßen die Lernenden dabei auf das Problem, dass als Ergebnis für die logarithmierte Modellwahrscheinlichkeit minus unendlich ausgegeben wird. Sie erhalten die Aufgabe, das Zustandekommen dieses Ergebnisses zu erklären. Durch gezielte Hinweise werden sie zunächst dazu aufgefordert, am Graph der Logarithmusfunktion zu untersuchen, für welche x -Werte die Logarithmusfunktion sehr große negative Werte annimmt und daraus Schlussfolgerungen auf den Wert der Modellwahrscheinlichkeit zu ziehen. Das Grenzverhalten

von Funktionen wird eigentlich erst in der Kursstufe untersucht (Kultusministerkonferenz, 2012, S. 18). Durch die graphische Darstellung können Lernende allerdings auch ohne das Konzept des Grenzwertes zu kennen, verstehen, dass der Logarithmus immer größere negative Werte annimmt, je näher die x -Werte der null kommen.

Anschließend sollen sie überlegen, welche Übergangswahrscheinlichkeit einem Wort unter dem interpolierten Tri-Gramm-Modell zugewiesen wird, wenn dieses Wort nicht im Trainingsdatensatz aufgetreten ist. Dadurch wird das Problem des bisherigen Sprachmodells entdeckt und die Notwendigkeit einer Glättung motiviert.

Auf Basis der Information, dass eine Gleichverteilung auf dem Vokabular $V = 75000$ angenommen wird, bestimmen die Lernenden die Wahrscheinlichkeit eines einzelnen Wortes zu $\frac{1}{V}$. Im Anschluss werden die logarithmierte Modellwahrscheinlichkeit unter dem geglätteten interpolierten Tri-Gramm-Modell und die zugehörige Kreuzentropie ermittelt. Da die Ergebnisse für sich genommen wenig aussagekräftig sind, werden Vergleiche mit anderen Sprachmodellen angestellt.

Vergleich verschiedener Modelle anhand der Kreuzentropie

Das geglättete interpolierte Tri-Gramm-Modell wird zuerst mit einem Sprachmodell verglichen, das Wörter aus einem vorgegebenen Vokabular V ohne ein zugrunde liegendes System rein zufällig vorschlägt. Man erwartet demnach, dass dieses Modell deutlich schlechter als das interpolierte Tri-Gramm-Modell abschneidet. Um den blinden Zufall zu modellieren, wird eine Gleichverteilung auf dem Vokabular V angenommen. Erneut sollen die Lernenden die Wahrscheinlichkeit für das Auftreten eines Wortes berechnen, die sich zu $\frac{1}{V}$ ergibt. Anschließend wird für dieses Zufalls-Modell die Kreuzentropie bestimmt, indem einfache Termumformungen durch Anwenden der Logarithmusgesetze durchgeführt werden:

$$\begin{aligned} K_T(P_{1,0,0,0}) &= -\frac{1}{n} \cdot \log P_{1,0,0,0}(X_1 = w_1, \dots, X_n = w_n) \\ &= -\frac{1}{n} \cdot \log \left(\left(\frac{1}{V} \right)^n \right) \\ &= -\frac{1}{n} \cdot n \cdot \log \left(\frac{1}{V} \right) = -\log \frac{1}{V} = \log V. \end{aligned}$$

Hier steht Unterstützung in Form zahlreicher gestufter Tipps zur Verfügung (vgl. Anhang A.1). Der zugehörige Funktionswert des Logarithmus kann dann über die Eingabe im Code bestimmt werden. Aus dem Vergleich wird deutlich, dass die Kreuzentropie des geglätteten interpolierten Tri-Gramm-Modells deutlich geringer ist als die des Zufalls-Modells (vgl. Tabelle 12.1). Dies zeigt, dass das entwickelte Sprachmodell für diese Art von Texten wesentlich bessere Vorhersagen trifft als ein Modell, das Wörter nur zufällig auswählt – ein Ergebnis, das aufgrund der strukturierten Natur der Sprache zu erwarten war.

Zudem sollen das geglättete interpolierte Tri-Gramm-Modell anhand der Kreuzentropie mit dem geglätteten Uni-, Bi- und Tri-Gramm-Modell verglichen werden und daraus Schlussfolgerungen für deren Eignung in der Anwendung gezogen werden (vgl. Kapitel 12). Anschließend nutzen die Lernenden diese Erkenntnisse, um die Gewichte für das geglättete interpolierte Tri-Gramm-Modell festzulegen. Bereits an

dieser Stelle wird das Ausprobieren verschiedener Gewichtungen als mögliche Strategie für das Ermitteln möglichst sinnvoller Gewichte angeregt. In der anschließenden Besprechung können die verschiedenen Gewichtungen und die dazugehörigen Kreuzentropiewerte verglichen werden. Aus diesem Vergleich ergibt sich ganz automatisch die Frage: „Was ist diejenige Gewichtung, für die die Kreuzentropie minimal wird?“.

13.2.6 Arbeitsblatt 5: Festlegen der Interpolationsgewichte und Diskussion

⑩ Einordnung in den Modellierungskreislauf

Auf Arbeitsblatt 4 wurde die Kreuzentropie als Qualitätsmaß eingeführt und ihr Wert für Modelle mit verschiedenen Interpolationsgewichten berechnet. Sie soll auf Arbeitsblatt 5 als Fehlerfunktion eingesetzt werden, um die optimalen Gewichte unter diesem Qualitätsmaß zu ermitteln (vgl. Anhang A.5). Das Schätzen der Interpolationsgewichte kann dann gegenüber dem heuristischen Festlegen der Gewichtung als Modellverbesserung aufgefasst werden. Zudem erfolgt eine letzte Gesamtbetrachtung von N -Gramm-Modellen, in der Grenzen und Chancen des Modells gesammelt werden und eine kritische Diskussion zu ethischen Aspekten hinsichtlich der Entwicklung und dem Einsatz von Sprachmodellen geführt wird.

Festlegen der Interpolationsgewichte

Die Lernenden haben die Möglichkeit, in einer längeren Version dieses Arbeitsblatts selbstständig ein Optimierungsverfahren zur Minimierung der Kreuzentropie in Abhängigkeit der Gewichte zu entwickeln (vgl. Abschnitt 13.2.7) oder alternativ – und stärker angeleitet – auf einer kürzeren Version des Arbeitsblatts, lediglich die Nebenbedingungen und die Zielfunktion des Optimierungsproblems aufzustellen. Die Optimierung wird dann als Black Box durchgeführt. Die ermittelte numerische Lösung soll anschließend mit den heuristischen Vorüberlegungen zur Gewichtung verglichen werden. Sie stimmt insofern mit den Vorüberlegungen überein, als dass auch hier das Gewicht des Tri-Gramm-Modells am größten und das Gewicht des Modells der Gleichverteilung am kleinsten gewählt wird.

Dieses Problem wird in der Besprechungsphase insofern als Schätzproblem erörtert, als dass hervorgehoben wird, dass die optimale Gewichtung ausschließlich für den vorliegenden Datensatz bestimmt wurde. Dieser stellt – wie der Trainingsdatensatz, der zur Schätzung der Übergangswahrscheinlichkeiten genutzt wurde – nur eine Stichprobe der Grundgesamtheit aller Texte dar. Wird also angenommen, dass diese Parameter im Allgemeinen eine sinnvolle Wahl für das Sprachmodell sind, so werden Schlussfolgerungen von einer Stichprobe auf die Grundgesamtheit gezogen. Das macht nur Sinn, wenn der Datensatz als repräsentativ angesehen werden kann. Dieser Aspekt kann mit den Lernenden später im Rahmen der Grenzen des Modells kritisch hinterfragt werden. Eine formalisierte Darstellung des Vorgehens als Schätzproblem ist aufgrund des begrenzten Vorwissens der Lernenden und aufgrund die zur Verfügung stehenden begrenzten Zeit, nicht sinnvoll. In der Besprechung kann darauf eingegangen werden, dass mit dem Minimieren der Kreuzentropie in

Abhängigkeit der Gewichte gleichzeitig ein Maximum der Modellwahrscheinlichkeit ermittelt wird und somit diejenigen Gewichte ermittelt werden, unter welchen der Datensatz die größte Auftretenswahrscheinlichkeit besitzt. Dadurch kann zumindest die Idee des Maximum-Likelihood-Prinzips verdeutlicht werden.

Zusätzlich muss der Hinweis erfolgen, dass für einen Vergleich der Qualität der verschiedenen Sprachmodelle die Bewertung des Sprachmodells auf einem unbekannten Datensatz erfolgen muss. Dieser darf noch nicht genutzt worden sein, um Parameter des Modells festzulegen. Die Bewertung auf einem unbekannten Datensatz findet im Lernmaterial allerdings nicht, was bereits Kapitel 12 dargelegt und begründet wird.

Diskussion

Abschließend werden zusammen mit den Lernenden Grenzen des Modells gesammelt, die bereits in Abschnitt 11.6 dargelegt werden. Im Plenumsgespräch werden insbesondere die begrenzte Erfassung des Kontextes im Modell und die fehlende Berücksichtigung grammatischer Regeln genannt. Zudem wird thematisiert, dass der Wandel der Sprache im Modell nicht abgebildet werden kann (vgl. Anhang B.7).

An den Workshop schließen sich gesellschaftliche Fragestellungen zu ethisch bedenklichen oder problematischen Aspekten bei der Nutzung und Entwicklung von Sprachmodellen an, diskutiert werden können. In Kleingruppen können zunächst Meinungen ausgetauscht und auf Kärtchen oder auf einer digitalen Pinnwand gesammelt werden. Anschließend werden die Antworten gemeinsam geclustert, und sie können durch den Input der Betreuungsperson mithilfe der Präsentationsfolien (siehe A.10) ergänzt werden. Es können Aspekte wie Beeinflussung des Sprachverhaltens durch Assistenzsysteme bei der Texteingabe (zum Beispiel beim Schreiben von Restaurantkritiken (Arnold et al., 2018)), Diskriminierung durch Sprachmodelle (zum Beispiel im Fall vom Chat Bot Tay von Microsoft (Arnold et al., 2018)), Ressourcenverbrauch beim Training eines Sprachmodells (zum Beispiel der Ressourcenverbrauch beim Training von GPT-3 (Patterson et al., 2021, S. 7)) diskutiert werden. Durch das Wissen um die Funktionsweise der Modelle kann von den Lernenden insbesondere für die letzten beiden genannten Aspekte leichter nachvollzogen werden, weshalb diese Zustände kommen. Vorurteilsbehaftete Wortvorschläge entstehen beispielsweise durch vorurteilsbehaftete Trainingsdaten. Durch den Einsatz von Sprachmodellen zur Textgenerierung – sei es durch einfache Wortvorschläge oder durch generative Systeme, die vollständige Texte erstellen – werden bestehende Vorurteile, die in Trainingsdaten enthalten sind, weiterverbreitet und können dadurch verstärkt werden (Bender et al., 2021, S. 617).

Die Beschäftigung mit diesen Fragestellungen dient somit der Aufklärung über Gefahren von KI-Systemen am Beispiel des hier entwickelten Systems, zum anderen zeigt sie einen zweiten Blickwinkel aus gesellschaftswissenschaftlicher Sicht auf das Problem und trägt durch die kritische Beleuchtung von Sprachmodellen zur Befähigung der Lernenden zu selbständiger kritischer Urteilsbildung und eigenverantwortlichem, bewussten Handeln bei.

13.2.7 Weiterführendes Material

Der Modellierungstag kann bei mehr Zeit um weitere Arbeitsphasen ergänzt werden. Es können zum einen im Standardmaterial als Blackbox eingeführte Verfahren wie das Optimierungsverfahren näher betrachtet werden. Zum anderen können weiterführende Aspekte des Themas wie die Rolle der Qualität der Trainingsdaten, die Rückfallstrategie und die Nutzung des Sprachmodells zur Textgenerierung bearbeitet werden. Dabei handelt es sich auch um zwei Teilprobleme, welche den Lernenden ohne kleinschrittige Aufgabenstruktur als offene Aufgabe gegeben werden. Bei der Bearbeitung der beiden Teilprobleme sind entweder mehr Programmiererfahrung oder eine intensivere Betreuung notwendig. Die Zusatzblätter sind zudem als differenzierende Maßnahme gedacht, um leistungsstarke und motivierte Schülerinnen und Schüler zu fördern (vgl. Abschnitt 8.1.3).

An dieser Stelle werden ausschließlich diejenigen Zusatzblätter vorgestellt, die entweder neue mathematische Inhalte einführen oder bereits angesprochene mathematische Inhalte vertiefend behandeln. Zusatzblätter, die sich ausschließlich mit der praktischen Umsetzung bereits erarbeiteter mathematischer Prozesse im Programmcode befassen, bleiben hier unberücksichtigt⁴.

Optimierungsverfahren

In einer erweiterten Version des fünften Arbeitsblatts wird im Unterschied zur kürzeren Version die Entwicklung eines Optimierungsverfahrens als offene Aufgabe gestellt (vgl. Anhang A.6). Das Arbeitsblatt kann daher entweder direkt als Alternative zu Arbeitsblatt 5 eingesetzt, oder leistungsstarken Lernenden als weiterführende tiefergehende Auseinandersetzung mit dem vorliegenden Optimierungsproblem nach der Bearbeitung des Arbeitsblattes 5 als Zusatzblatt gegeben werden.

Bei der Entwicklung des Optimierungsverfahrens darf auf begleitende Tipps zurückgegriffen werden. Die Tipps weisen darauf hin, verschiedene Gewichtungen systematisch zu erproben und die zugehörigen Kreuzentropiewerte zu bestimmen. Dieser Ansatz führt zu einer vollständigen Suche im Suchraum – eine Methode, die für dieses Optimierungsproblem am ehesten von den Lernenden selbstständig erarbeitet werden kann. Es wird auch als Tipp gegeben, das Optimierungsproblem wie in Abschnitt 11.5.2 vorgeschlagen umzuformulieren, sodass das Minimum der Zielfunktion

$$\widetilde{K}^*(g_0, g_1, g_2) := \widetilde{K}(g_0, g_1, g_2, 1 - g_0 - g_1 - g_2)$$

auf $M^* = \{(g_0, g_1, g_2) \in \mathbb{R}^3 : 0 \leq g_0, g_1, g_2 \leq 1\}$ gesucht ist. Das hat den Vorteil, dass der Suchraum dann im dreidimensionalen Anschauungsbereich liegt und eine Diskretisierung zum Beispiel als quadratisches Gitternetz vorstellbar ist. Die Formulierung des Optimierungsproblems, die Wahl des Optimierungsverfahrens sowie die konkrete Diskretisierung werden jedoch von den Lernenden selbstständig erarbeitet.

⁴Ein Beispiel hierfür ist die Automatisierung der Berechnung von Übergangswahrscheinlichkeiten im Bi-Gramm-Modell, bei der die absoluten Häufigkeiten einzelner Wortsequenzen in Tabellen zwischengespeichert werden.

Die Tipps unterstützen daher nur dann, wenn sie zum gewählten Ansatz passen. Andernfalls kann eine individuelle Hilfestellung durch die Betreuungsperson notwendig sein. Es ist zum Beispiel auch möglich, die ursprüngliche Formulierung des Optimierungsproblems beizubehalten und die optimale Gewichtung im diskretisierten ursprünglichen Suchraum zu bestimmen, der die Form eines 4-Simplexes hat. Nach der Bestimmung der optimalen Gewichtung soll diese, wie bereits in Arbeitsblatt 5, mit den vorangegangenen heuristischen Überlegungen verglichen werden. Die Ergebnisse sowie das entwickelte Optimierungsverfahren werden von den Lernenden im Plenum präsentiert und die verschiedenen Verfahren miteinander diskutiert.

Über die vollständige Suche wird zwar eine optimale Lösung gefunden, allerdings nur innerhalb der gewählten Diskretisierung des Problems. Aufgrund der hohen Rechenzeit für die Berechnung einzelner Werte der Kreuzentropie kann die Diskretisierung nicht allzu feingranular erfolgen. Für die Zwecke des Lernmaterials ist dies jedoch ausreichend. Der Zusammenhang zwischen Rechenzeit und Feinheit der Diskretisierung kann bei der Präsentation der Optimierungsverfahren thematisiert und als Vergleichskriterium herangezogen werden.

Textgenerierung mit Wortvorschlägen und ein Vergleich mit ChatGPT

In dieser Zusatzaufgabe (vgl. Anhang A.7) soll das entwickelte *N*-Gramm-Modell genutzt werden, um nicht nur einzelne Wörter vorherzusagen, sondern ganze Texte zu erzeugen. Das Zusatzblatt ist an verschiedenen Stellen im Lernmaterial verlinkt. Es kann ab der abgeschlossenen Bearbeitung des dritten Arbeitsblattes sinnvoll bearbeitet werden.

Da hier das Sprachmodell nicht nur zum Vorschlagen einzelner Wörter, sondern zum Erzeugen ganzer Texte genutzt wird wird wie von Pohjolainen und Heiliö (2016, S. 4) für gutes Material zur Modellierung vorgeschlagen veranschaulicht, dass ein entwickeltes Modell häufig vielseitig einsetzbar ist. Die Lernenden erhalten lediglich die Information, dass sie eines der entwickelten Modelle (das Uni-, Bi-, das Tri-Gramm-Modell, das interpolierte Tri-Gramm-Modell, Tri-Gramm-Modell mit Rückfallstrategie) zur Texterzeugung nutzen sollen. Das Ziel der Aufgabe ist es, ähnlich wie im Unterrichtsvorschlag von Oldenburg (2008) eine Wortsequenz als Startsequenz vorzugeben und anschließend dem Text immer wieder den nächsten Wortvorschlag anzuhängen. Bei Bedarf erhalten die Lernenden den Tipp, dass sie in einer Nachrichtenapp ihrer Wahl mehrmals hintereinander den mittleren Wortvorschlag auswählen sollen.

Es fällt schnell auf, dass zwar stückweise Textpassagen erzeugt werden, die sinnhaft klingen, aber einige davon keinen Sinn ergeben. Es werden teilweise semantisch und/ oder grammatisch inkorrekte Textpassagen generiert. Auch die fehlenden Satzzeichen und Endlosschleifen sind ein Problem. In einer zweiten Aufgabe werden die Schülerinnen und Schüler daher aufgefordert, die aufgetretenen Probleme zu beschreiben und Ergebnisse verschiedener Modelle zu vergleichen.

Es fällt auf, dass das Uni-Gramm-Modell immer wieder das gleiche wahrscheinlichste Wort ausgibt und daher gänzlich ungeeignet ist. Die vom Bi-Gramm-Modell erzeug-

ten Texte sind nur wenig besser. Das Auftreten allein vom vorangegangenen Wort abhängig zu machen erweist sich als ungenügend. Das Tri-Gramm-Modell schneidet im Vergleich besser ab. Bei beiden Modellen, aber insbesondere beim Tri-Gramm-Modell kann es jedoch passieren, dass die Startsequenz nicht im Trainingsdatensatz auftritt und daher überhaupt kein Text generiert werden kann. Die besten Texte liefern das interpolierte Tri-Gramm-Modell und das Tri-Gramm-Modell mit Rückfallstrategie. Insgesamt können die folgenden Probleme zusammengefasst und mögliche Verbesserungen gesammelt werden:

- Dieselbe eingegebene Wortvorgeschichte liefert aufgrund eines eindeutig wahrscheinlichsten Folgewortes immer denselben Output. Deshalb kann es bei der Generierung zu Endlosschleifen kommen. Statt lediglich das wahrscheinlichste Wort auszuwählen, kann das Auswählen des nächsten Wortes als Zufallsexperiment stattfinden, bei dem die Wahrscheinlichkeit für das Auswählen eines Wortes über dessen Übergangswahrscheinlichkeiten gegeben ist. Dies sorgt für mehr Variation. Alternativ kann aus den drei wahrscheinlichsten Wörtern zufällig eines als Folgewort ausgewählt werden.
- Im neu generierten Text kommen lediglich aus dem Trainingstext bekannte Wortformen und Wortsequenzen vor. Bei fortschrittlicheren Sprachmodellen werden statt eines ganzen Wortes deshalb kleinere Einheiten, wie beispielsweise der Wortstamm oder einzelne Silben eines Wortes betrachtet. Gleichzeitig ist der Trainingsdatensatz um einiges größer als der hier verwendete.
- Das Auftreten allein von wenigen vorangegangenen Wörtern abhängig zu machen ist nicht ausreichend für passable Textergebnisse. Der Kontextbezug könnte erhöht werden, indem längere Wortvorgeschichten betrachtet werden. Stehen allerdings Wörter in einem Satz des Trainingstextes in Beziehung, die mehr als nur wenige Worte voneinander entfernt sind, so kann deren Beziehung durch das N -Gramm-Modell nicht erfasst werden. Die Berechnung der N -Gramm-Wahrscheinlichkeit würde bei großem N aufgrund geringer Auftretenshäufigkeiten der N -Gramme gar nicht möglich sein oder nur sehr ungenau werden. Es wäre deshalb weder praktikabel noch sinnvoll, Folgen von mehr als fünf Wörtern in den bedingten Wahrscheinlichkeiten zu berücksichtigen.
- Satzzeichen werden im bisherigen Sprachmodell nicht berücksichtigt, da sie für die Nutzung des Sprachmodells zur Wortvorhersage nicht weniger sind. Dies führt dazu, dass sie im generierten Text manuell gesetzt werden müssen. Eine Aufnahme der Satzzeichen als Zustände oder zumindest der Erfassung der Satzenden beim Training würde dem entgegenwirken.

Wird die Zusatzaufgabe allen Lernenden als weiterführende Aufgabe gegeben, so kann ein Vergleich mit populären generativen KI-Systemen wie ChatGPT im Unterrichtsgespräch thematisiert werden. Das Sprachmodell von ChatGPT, bekannt als GPT (Generative Pre-trained Transformer), weist einige grundlegende Gemeinsamkeiten mit dem N -Gramm-Modell auf. Beide basieren auf statistischen Zusammenhängen, die aus Worthäufigkeiten in Textdaten abgeleitet werden. Sie generieren Texte, indem Wörter nach einer durch das Sprachmodell vorgegebenen Wahrscheinlichkeitsverteilung zusammengesetzt werden. GPT implementiert jedoch wesentlich

komplexere Verfahren, die über die Ansätze des *N*-Gramm-Modells hinausgehen (Bender et al., 2021, S. 616).

Während das *N*-Gramm-Modell lediglich kurze aufeinanderfolgende Wortsequenzen sequentiell von links nach rechts analysiert, verarbeitet GPT den gesamten Text als Einheit. Dies ermöglicht es, auch semantische und syntaktische Beziehungen zwischen Wörtern in einem Text, unabhängig von deren Abstand zueinander, zu erkennen (Vaswani et al., 2017, S. 3 - 5). Diese globale Textanalyse erlaubt es, Kontexte besser zu erkennen, was für die Qualität der generierten Texte entscheidend ist.

Trotz der fortschrittlichen Architektur von GPT ist die primäre Funktion nach der Trainingsphase zunächst auf die Vorhersage der nächsten Wortsilben oder Wörter beschränkt. Bessere Ergebnisse werden erst durch eine Feinabstimmung erreicht, bei der menschliches Feedback systematisch in den Trainingsprozess integriert wird (Stennnon et al., 2020, S. 1). Trotz der dadurch erreichbaren sprachlichen Eloquenz des Systems muss klar sein, dass es die semantische Bedeutung des produzierten Textes nicht versteht, da es kein Weltwissen oder Bewusstsein besitzt. Stattdessen handelt es sich wie auch beim *N*-Gramm-Modell um „stochastische Papageien“ (Bender et al., 2021, S. 617), die lediglich einzelne sprachliche Elemente aus Trainingstexten auf der Grundlage probabilistischer Informationen neu kombinieren.

Rolle der Qualität der Trainingsdaten

Auf einem weiteren optionalen Arbeitsblatt (vgl. Anhang A.8) soll das aus der Statistik bekannte Problem nicht-repräsentativer Stichproben thematisiert werden. Dieses Problem stellt eine zentrale Herausforderung für die Entwicklung und Anwendung von KI-Systemen dar. Wenn die Trainingsdaten keine repräsentative Stichprobe der Grundgesamtheit sind, können die Vorhersagen verzerrt ausfallen (Botsch, 2023, S. 8). Selbst wenn die Daten ursprünglich repräsentativ waren, können Veränderungen der Bedingungen in der Zukunft die Prognosen unzuverlässig machen. Dieses Problem zeigt sich am Beispiel der Wortvorhersage, wenn die Schätzung von Wahrscheinlichkeiten auf Basis relativer Häufigkeiten aus der Vergangenheit für die Vorhersage zukünftiger Ereignisse genutzt werden. Um das Verständnis für die Problematik verzerrter Stichproben bei der Nutzung von KI zu vertiefen, wird dieser Aspekt gemeinsam mit den Lernenden an einem Beispiel erarbeitet.

Das Arbeitsblatt ist als weiterführendes Zusatzmaterial für schnelle und motivierte Lernende am Ende von Arbeitsblatt 4 und Arbeitsblatt 5 verlinkt. Alternativ kann es anstelle des vierten Arbeitsblattes nach der Einführung der Kreuzentropie von allen im Plenum oder im Anschluss an das fünfte Arbeitsblatt bearbeitet werden.

Um die Rolle der Qualität der Trainingsdaten zu untersuchen, wird ein zweiter Datensatz, bestehend aus Werken von Johann Wolfgang von Goethe (1774a, 1774b, 1796, 1809) mit rund 150.000 Wörtern, für das Training des mit den Lernenden erarbeiteten Sprachmodells verwendet. Dieses neue Modell sowie das mit dem ursprünglichen Datensatz trainierte Sprachmodell sollen durch die Eingabe unterschiedlicher Wortvorgeschichten erprobt und die ausgegebenen Vorschläge verglichen werden.

Dabei fällt auf, dass teilweise Wörter vorgeschlagen werden, die heute nicht mehr gebräuchlich sind und/ oder als wenig passend für Textnachrichten erscheinen.

Anschließend soll die Güte der beiden mit unterschiedlichen Daten trainierten Modelle verglichen werden. Es zeigt sich, dass das mit den Texten von Goethe trainierte Sprachmodell eine deutlich höhere Kreuzentropie als das mit dem ursprünglichen Datensatz trainierte Sprachmodell aufweist.

Zum Schluss sollen allgemeine Anforderungen an die Eigenschaften eines Trainingsdatensatzes formuliert werden. Ziel ist es, die Bedeutung eines geeigneten Trainingsdatensatzes für ML-Methoden herauszuarbeiten. Im Fall des Sprachmodells muss der Datensatz die Anwendungssituation hinsichtlich des Vokabulars und der Textart passend repräsentieren (Pfister & Kaufmann, 2017, S. 387). Wird das Sprachmodell wie hier zur Wortvorhersage beim Chatten genutzt, so sollten die Trainingstexte demnach nicht unbedingt aus dem 17. Jahrhundert stammen, da damals andere Wörter als in der heutigen Gegenwartssprache verwendet wurden und dadurch natürlich auch für die Anwendung weniger passende Vorschläge generiert werden.

Rückfallstrategie

Dieses Zusatzblatt kann von besonders schnellen und motivierten Lernenden nach der Bearbeitung von Arbeitsblatt 3 bearbeitet werden. Auf diesem Arbeitsblatt wird die Rückfallstrategie erarbeitet, die in Abschnitt 11.3 als eine der typischen Methoden zur Kombination der Modelle vorgestellt wird (vgl. Anhang A.7). Um die Methode möglichst einfach zugänglich zu machen, wird sie in diesem Zusatzblatt in einer Abwandlung erarbeitet. Statt einen kombinierten Schätzer für die Übergangswahrscheinlichkeiten zu ermitteln, wird je nach Situation ein anderes Modell verwendet, um das nächste Wort vorherzusagen. So kann die Renormierung der Wahrscheinlichkeiten umgangen werden, ohne stochastisch inkorrekte Aussagen zu treffen. Die Lernenden sollen sich zunächst das folgende Szenario vorstellen. Es liegt eine Wortvorgeschichte der Länge zwei vor, die nicht im Datensatz auftritt. Es kann daher mit dem Tri-Gramm-Modell kein Vorschlag gemacht werden. Es soll nun für diesen Fall eine Strategie gefunden werden, wie unter Rückbezug der anderen Modelle dennoch ein Vorschlag ausgegeben werden kann. Darauf aufbauend soll eine allgemeine Strategie formuliert werden, mit der es möglich ist, die Wortvorgeschichte so lange wie möglich zu wählen und dennoch immer einen Vorschlag zu generieren.

Im Anschluss wird erklärt, weshalb die Interpolation im Vergleich zu der auf diesem Arbeitsblatt präsentierten Methode die bessere Möglichkeit ist. Da vor allem die Tri-Gramm- und manchmal auch die Bi-Gramm-Wahrscheinlichkeiten häufig auf sehr wenigen Daten beruhen, ist zu befürchten, dass diese oft nicht zuverlässig sind. Hierzu wird im Lernmaterial ein Beispiel gemacht: Das Tri-Gramm-Modell liefert drei mögliche Vorschläge. Hierbei kann es passieren, dass die zugehörigen Tri-Gramme im Datensatz nur ein einziges Mal auftreten. Die Vorschläge basieren dann auf sehr geringen Auftretenshäufigkeiten und bilden nicht unbedingt den typischen Satzverlauf zu der gegebenen Wortvorgeschichte ab. In diesem Fall ist es sinnvoll, einen zweiten Blick auf die aus den Bi-Gramm-Wahrscheinlichkeiten generierten Vorschläge, zu

werfen. Zählen die Vorschläge des Tri-Gramm-Modells auch hier zu den besten Vorschlägen dazu, so kann man sich sicherer sein, dass es sich bei den Wortvorschlägen aus dem Tri-Gramm-Modell tatsächlich um Wörter handelt, die in diesem Kontext häufiger verwendet werden. Treten die Wortvorschläge des Tri-Gramm-Modells im Bi-Gramm-Modell mit einer im Vergleich zu anderen Wörtern eher geringen Wahrscheinlichkeit auf, so sollte man abwägen, welche der Wörter sowohl bei der Bi- als auch der Tri-Gramm-Wahrscheinlichkeit insgesamt die größte Wahrscheinlichkeit bei gegebener Wortvorgeschichte haben.

Es ist daher sinnvoll, die Übergangswahrscheinlichkeiten des Bi-Gramm-Modells bei der Bestimmung eines Wortvorschlages mit zu berücksichtigen. Auch die Information, wie häufig ein Wort im Allgemeinen auftritt, kann entscheidend für die Wahl des Wortvorschlages sein. Das Uni-Gramm-Modell sollte daher ebenfalls zur Bildung der Übergangswahrscheinlichkeit beitragen. Dies kann durch die Verwendung des interpolierten Tri-Gramm-Modells garantiert werden.

13.2.8 Exkurs: Umsetzung in einer Doppelstunde

Aufgrund der in Abschnitt 9.3 dargelegten Vorteile eines kompakteren Formats für den Schulunterricht wird zusätzlich eine Doppelstunde zum Thema Wortvorhersage konzipiert. Die Doppelstunde orientiert sich inhaltlich weitgehend an der Einführungs- sowie der ersten Erarbeitungs- und Sicherungsphase des Tagesworkshops. Das Ziel der Unterrichtseinheit ist die Erarbeitung des Bi-Gramm-Modells zur Wortvorhersage aufgrund von zentralen Inhalten des Stochastikunterrichts bis Jahrgangsstufe 9.

Zu Beginn erfolgt eine Einführung in den Anwendungskontext der Verarbeitung natürlicher Sprache als Teilbereich von KI. Die Funktionsweise des Bi-Gramm Modells wird wie im Tagesworkshop in Partnerarbeit an einem ersten kleinen Beispiel erarbeitet. Ein bereits vorgezeichneter Übergangsgraph wird auf Grundlage eines kleinen Textdatensatzes mit den zugehörigen relativen Häufigkeiten von den Lernenden vervollständigt. Im Unterschied zum Tagesworkshop wird zusätzlich ein weiteres Beispiel bearbeitet: Die Lernenden erstellen eigenständig einen Übergangsgraphen und eine Übergangstabelle für einen zweiten, etwas umfangreicheren Trainingsdatensatz. Dabei bestimmen sie sowohl die Knotenpunkte als auch die relativen Häufigkeiten. Dieses zweite Beispiel dient einerseits der Übung in der Berechnung von relativen Häufigkeiten und der Erstellung von Übergangsgraphen. Andererseits verdeutlicht es aufgrund des größeren Trainingsdatensatzes, dass die Berechnung der relativen Häufigkeiten mit wachsender Textlänge zunehmend komplex wird. Diese Erkenntnis motiviert die Automatisierung des Berechnungsprozesses und das Aufstellen einer entsprechenden Formel für die Berechnung der relativen Häufigkeiten. Im Tagesworkshop wird der Datensatz aus dem ersten Beispiel hingegen nur erweitert und hierfür der Übergangsgraph erstellt, da die Erstellung eines Übergangsgraphen auch beim Tri-Gramm-Modell geübt wird (vgl. Abschnitt 13.2.3).

Auch die methodische Umsetzung wird an das kürzere Format angepasst. Während im Tagesworkshop eine digitale Lernumgebung auf Basis von Jupyter Notebooks

zum Einsatz kommt, wird in der Doppelstunde bewusst auf analoge Lernmaterialien gesetzt. Diese Entscheidung wird getroffen, um die Einarbeitungszeit einzusparen, die Lernende benötigen, um sich mit den digitalen Arbeitsblättern vertraut zu machen. Die Einführung in den Umgang mit Jupyter Notebooks kann zwar sehr kurz gehalten werden, dennoch ist zu beobachten, dass zunächst etwas Zeit benötigt wird, um mit den digitalen Arbeitsblättern zurecht zu kommen und sich dies auch in den Bearbeitungszeiten insbesondere des ersten Arbeitsblatts zeigt. In einem Tagesworkshop ist dieser Aufwand vertretbar, da die Gesamtzeit genügend Spielraum bietet. In einer Doppelstunde hingegen würde der Anteil der Einarbeitungszeit im Verhältnis zur gesamten Unterrichtszeit verhältnismäßig hoch ausfallen.

Da jedoch große Datenmengen für die Entwicklung eines tatsächlich einsatzfähigen *N*-Gramm-Modells verarbeitet werden müssen, ist ein vollständiger Verzicht auf digitale Hilfsmittel nicht praktikabel. Als Kompromiss wird die Doppelstunde daher so konzipiert, dass die Entwicklung des *N*-Gramm-Modells auf der Basis eines kleinen Textdatensatzes zunächst mithilfe analoger Arbeitsblätter geschieht. Anschließend kann das auf einem größeren Datensatz trainierte Modell in einer Webanwendung erprobt werden (vgl. Abbildung 13.7). In dieser Anwendung können die Lernenden verschiedene Wortvorgeschichten eingeben und erhalten aufgrund der errechneten relativen Häufigkeiten im Trainingsdatensatz Vorschläge für das nächste Wort. Diejenigen drei Wörter, die auf die Wortvorgeschichte mit der größten Wahrscheinlichkeit folgen, werden als Vorschläge angezeigt.

Workshop: Wortvorschläge

In dieser Webapplikation kannst du das Bi-Gramm-Modell anhand eines Trainingsdatensatzes*, der **über 300.000 Wörter** umfasst, erproben. Gib dazu verschiedene Wortvorgeschichten ein. Du erhältst über das auf deinen Überlegungen basierende Verfahren **drei Wortvorschläge**.

*Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"¹ und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"².

¹ Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

² Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

Abbildung 13.7: Bildschirmfoto der Webanwendung (<https://workshops.cammp.online/demo>)

Beim Ausprobieren des Bi-Gramm-Modells können die Lernenden die Möglichkeiten und Grenzen des Modells erkunden. Ihre Erkenntnisse werden dann ähnlich wie im Tagesworkshop im Plenum gesammelt. Anschließend werden im Plenum Verbesserungsideen diskutiert, wobei ein Ausblick auf das Uni- und Tri-Gramm-Modell gegeben wird. Den Abschluss bildet eine Diskussion mit gesellschaftskritischer Perspektive zu den Auswirkungen und Eigenschaften von Sprachmodellen, analog zum Modellierungstag.

In einer Zusatzaufgabe kann das entwickelte Bi-Gramm-Modell, ähnlich wie für den Tagesworkshop in Abschnitt 13.2.7 beschrieben, in einem begrenzten Rahmen ebenfalls zur Generierung von Texten eingesetzt und mit bekannten Sprachmodellen wie ChatGPT verglichen werden. Analog zum Unterrichtsvorschlag von Oldenburg (2008) wird ein Wort als Startwort vorgegeben, und anschließend wird dem Text immer wieder der nächste Wortvorschlag angehängt. Den Lernenden wird dazu die in Abbildung 13.8 gezeigte Übergangstabelle gegeben, mithilfe derer sie zu verschiedenen vorgegebenen Satzanfängen Sätze generieren sollen. Dabei wird schnell deutlich, dass die resultierenden Sätze nicht immer semantisch oder grammatisch korrekt sind. So entstehen beispielsweise Endlosschleifen wie „Wir fahren mit dem Auto fahren mit dem Auto fahren...“. Gleichzeitig können aber auch inhaltlich sinnvolle und grammatisch korrekte Sätze generiert werden, wie etwa „Oma nimmt den Bus, der in die Stadt fährt.“. Die anschließende Aufgabe ist es daher, die Probleme, die

		Wort2																		
		,	.	arbeitet	Auto	Bank	Bus	dem	den	der	die	fahren	fährt	in	Mama	mit	nimmt	Oma	Stadt	wir
Wort1	,	0	0	0	0	0	0	0	0,2	0,6	0,2	0	0	0	0	0	0	0	0	0
	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	arbeitet	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0	0,2	0	0	0	0
	Auto	0	0,3	0	0	0	0	0	0	0	0	0	0,5	0,2	0	0	0	0	0	0
	Bank	0,2	0,8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bus	0,6	0	0	0	0	0	0	0	0	0	0	0,4	0	0	0	0	0	0	0
	dem	0	0	0	0,4	0	0,3	0	0	0	0	0	0	0	0	0	0	0,3	0	0
	den	0	0	0	0	0	0,8	0	0	0	0	0	0	0	0	0	0	0,2	0	0
	der	0	0	0	0	0,2	0,2	0	0	0	0	0	0	0	0,3	0	0	0	0,1	0,2
	die	0	0	0	0	0,2	0	0	0	0	0	0	0	0	0,2	0	0	0	0,6	0
	fahren	0	0	0	0	0	0,3	0	0	0	0	0	0	0,2	0	0,5	0	0	0	0
	fährt	0	0,3	0	0,2	0	0,1	0	0	0	0	0	0	0,2	0	0,2	0	0	0	0
	in	0	0	0	0	0	0	0,1	0,3	0,2	0,4	0	0	0	0	0	0	0	0	0
	Mama	0	0	0,7	0	0	0	0	0	0	0	0	0,2	0	0	0	0,1	0	0	0
	mit	0	0,1	0	0	0	0	0,5	0	0,4	0	0	0	0	0	0	0	0	0	0
	nimmt	0	0	0	0	0	0	0	0,5	0	0,3	0	0	0	0,1	0,1	0	0	0	0
	Oma	0	0,1	0,1	0	0	0	0	0	0	0	0	0,3	0	0	0	0,5	0	0	0
	Stadt	0,1	0,4	0	0	0	0	0	0	0	0	0	0,5	0	0	0	0	0	0	0,1
	wir	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Abbildung 13.8: Übergangstabelle für die Generierung von Texten

bei der Textgenerierung auftreten, zu beschreiben. Falls das Zusatzblatt nicht von allen bearbeitet wird, so kann die Nutzung des Modells für die Textgenerierung von den Lernenden, die das Arbeitsblatt bearbeitet haben, kurz vorgestellt werden. Im Anschluss können die Probleme im Plenum gesammelt und mögliche Verbesserungen diskutiert werden. Die Probleme und Verbesserungsmöglichkeiten ähneln den Problemen, die bereits beim Zusatzblatt des Tagesworkshops erläutert wurden. Sie unterscheiden sich allerdings trotzdem in manchen Punkten aufgrund der Tatsache, dass im Unterrichtsansatz der Doppelstunde nur das Bi-Gramm-Modell getestet wird und Satzzeichen in der Übergangstabelle mit berücksichtigt sind. Die Probleme der Nutzung des Bi-Gramm-Modells zur Textgenerierung und mögliche Verbesserungen sollen deshalb hier kurz genannt werden.

- Dieselbe eingegebene Wortvorgeschichte liefert aufgrund eines eindeutig wahrscheinlichsten Folgewortes immer denselben Output. Deshalb kann es bei der Generierung zu Endlosschleifen kommen. Ein zufälliges Auswählen des nächsten Wortes anhand der über die Übergangswahrscheinlichkeiten vorgegebenen Verteilung oder aus den drei wahrscheinlichsten Folgewörtern sorgt für mehr Variation.
- Im neu generierten Text kommen lediglich aus dem Trainingstext bekannte Wortformen und Wortsequenzen vor. Bei fortschrittlicheren Sprachmodellen werden statt eines ganzen Wortes deshalb kleinere Einheiten, wie beispielsweise der Wortstamm oder einzelne Silben eines Wortes, betrachtet. Gleichzeitig ist der Trainingsdatensatz um einiges größer als der hier verwendete.
- Das Auftreten allein vom vorangegangenen Wort abhängig zu machen ist ungenügend. Stattdessen kann der Kontextbezug erhöht werden, indem mehr als ein Wort als Wortvorgeschichte betrachtet wird. Das Tri-Gramm-Modell kann hier als Verbesserung vorgestellt werden. Stehen allerdings Wörter in einem Satz des Trainingstextes in Beziehung, die mehr als nur wenige Worte voneinander entfernt sind, so kann durch das *N*-Gramm-Modell deren Beziehung nicht erfasst werden.

13.3 Curriculare Anbindung

Dieses Kapitel soll aufzeigen, dass das Lernmaterial eine Vielzahl an Inhalten integriert, die fest im schulischen Lehrplan verankert sind. Im Fokus steht dabei die curriculare Verknüpfung mit dem Bildungsplan für das Fach Mathematik in Baden-Württemberg (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024) sowie die Verknüpfung mit den bundesweit gültigen Bildungsstandards für den mittleren Schulabschluss im Fach Mathematik (Kultusministerkonferenz, 2022). Darüber hinaus werden Bezüge zu den Schulfächern Informatik und Deutsch aufgezeigt. Grundlage für diese Anbindung bilden ebenfalls die Bildungsstandards und der Bildungsplan Baden-Württembergs, da die Workshops hauptsächlich in diesem Bundesland durchgeführt wurden.

Anbindung an Curricula des Fachs Mathematik

Das Lernmaterial zur Wortvorhersage ist problembezogen konzipiert, sodass die im Material verwendeten mathematischen Inhalte zur Lösung des mathematischen Problems dienen und nicht um ihrer selbst willen eingeführt werden. Da es sich bei der Vorhersage des nächsten Wortes um ein reales Problem und nicht um eine konstruierte, auf zuvor gelernte Inhalte zugeschnittene Aufgabe handelt, hat dies zur Folge, dass Inhalte verschiedener Jahrgangsstufen und verschiedener mathematischer Bereiche zur Problemlösung notwendig sind und daher im Lernmaterial aufgegriffen werden. Das bietet die Möglichkeit, im Sinne des Spiralprinzips (Bruner & Hartung, 1973) bereits in verschiedenen Entwicklungsstufen behandelte Inhalte in einem neuen, komplexeren Kontext zu wiederholen und zu vertiefen.

Besonders viele Anknüpfungspunkte gibt es zu Kompetenzen der Leitidee *Daten und Zufall*. Dazu zählt der Umgang mit absoluten und relativen Häufigkeiten, der in Jahrgangsstufe 5/ 6 erlernt werden soll (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 25; Kultusministerkonferenz, 2022, S. 21). Um die relativen Häufigkeiten zu ermitteln werden Daten ausgewertet, indem die Anzahl gleicher Wortsequenzen im Trainingsdatensatz bestimmt wird. Zudem werden, wie in Jahrgangsstufe 7/8 im Bildungsplan verankert, „Aussagen, die auf einer Datenanalyse basieren, formulier[t] und bewerte[t]“ (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 32). In Jahrgangsstufe 7/ 8 sollen, wie es in Aufgabenblatt 1 und 2 am Beispiel der relativen Häufigkeiten der Wortsequenzen in Trainingstexten thematisiert wird, in Zufallsexperimenten beobachtete relative Häufigkeiten als Schätzwerte von Wahrscheinlichkeiten genutzt werden (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 32; Kultusministerkonferenz, 2022, S. 22). Bei der Erstellung von Übergangsgraph und Übergangstabelle arbeiten die Lernenden mit verschiedenen Darstellungsformen, lesen diese und interpretieren deren Einträge im Sachkontext. Auch hierbei werden Kompetenzen aufgegriffen und gefördert die speziell in der Leitidee *Daten und Zufall* (Kultusministerkonferenz, 2022, S. 21 - 22) aber auch unter den prozessbezogenen Kompetenzen genannt sind (Kultusministerkonferenz, 2022, S. 12). Auch auf Arbeitsblatt 4 wird an Kompetenzen der Leitidee *Daten und Zufall* angeknüpft. Hier stellen die Lernenden das mehrstufige Zufallsexperiment als Baumdiagramm dar und wenden die Pfadregeln zur Berechnung der Modellwahrscheinlichkeit an. Beide Kompetenzen sollen bereits in Jahrgangsstufe 7/8 erworben werden (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 33; Kultusministerkonferenz, 2022, S. 22).

Bei der Bestimmung der Kreuzentropie werden zudem Kompetenzen der Leitidee *Zahl und Operation* adressiert. Dazu zählen das Umformen von Termen (Jahrgangsstufe 7/ 8)(Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 26 - 27) auch unter Nutzung von Potenz- und Logarithmusgesetzen, die in Klassenstufe 9 thematisiert werden (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 34; Kultusministerkonferenz, 2022, S. 16). Der Graph der Logarithmusfunktion wird im Lernmaterial vorgegeben, weil das Material bereits ab Jahrgangsstufe 10 geeignet sein soll. Wird der Workshop allerdings erst in der Kursstufe durchgeführt, so besteht die Möglichkeit die Logarithmusfunktion von den Lernenden selbst skizzieren zu lassen. Das ist eine Kompetenz die auch im Bildungs-

plan in Jahrgangsstufe 11/ 12 verankert ist (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 47).

Wird in der längeren Version des fünften Arbeitsblattes die Lösung des Optimierungsproblems von den Lernenden numerisch bestimmt, so wird damit ein algorithmisches Verfahren entwickelt, was ebenfalls zu den benannten Kompetenzen der Leitidee *Zahl und Operation* zählt (Kultusministerkonferenz, 2022, S. 16). Die Kompetenz umfasst das „[W]ählen, [B]eschreiben und [B]ewerten [von] Vorgehensweisen und Verfahren, denen Algorithmen bzw. Kalküle zu Grunde liegen und [das] [Aus]führen diese[r] [...]“ (Kultusministerkonferenz, 2022, S. 16) sowie das „[I]mplementieren ein[es] algorithmische[n] Verfahren[s] [...] mit digitalen Mathematikwerkzeugen“ (Kultusministerkonferenz, 2022, S. 16).

Das Lernmaterial knüpft somit an das Vorwissen der Lernenden aus den Leitideen *Zahl und Operation* sowie *Daten und Zufall* bis einschließlich der 9. Jahrgangsstufe an. Aufgrund der Komplexität einiger mathematischer Methoden, die im Material zur Anwendung kommen, wird jedoch empfohlen, das Lernmaterial nicht bereits in der 9., sondern erst ab der 10. Jahrgangsstufe einzusetzen.

Das Material kann zudem als Ausgangspunkt für das Erarbeiten weiterführender Inhalte dienen. Die Beschreibung stochastischer Prozesse durch Übergangsmatrizen ist beispielsweise in manchen Bundesländern sogar explizit Teil des Lehrplans der Kursstufe (Kultusministerkonferenz, 2012, S. 18). Da die Darstellung eines stochastischen Prozesses als Übergangstabelle im Lernmaterial erarbeitet wird, kann das Material in diesen Bundesländern als einführendes Beispiel verwendet werden.

Anbindung an Curricula des Fachs Informatik

Auch der Lehrplan des Fachs Informatik bietet vielfältige Anknüpfungspunkte. Mit der geplanten Bildungsreform befindet sich das Fach Informatik aktuell im Umbruch. Das Auslaufen des Fachs IMP (Informatik, Mathematik, Physik) wird von der Einführung eines eigenständigen als „Informatik und Medienbildung“ bezeichneten Pflichtfachs begleitet, dessen Inhalte sich zum Teil an den bisherigen Informatikanteilen des Fachs IMP orientieren sollen. Ab dem Schuljahr 2025/26 soll dieses Pflichtfach durchgängig von Jahrgangsstufe 5 bis 11 eingeführt werden (Ministerium BW, 2024b). Das Thema Künstliche Intelligenz, was sich derzeit weder im Mathematik- noch im Informatik-Lehrplan explizit findet, soll in diesem Fach einer der Kerninhalte sein (Ministerium BW, 2023b). Das hier entwickelte Lernmaterial würde sich ideal für den Einsatz in diesem Fach eignen. Konkrete zugehörige Bildungspläne für das Fach Informatik und Medienbildung sind nach aktuellem Kenntnisstand noch nicht einsehbar, weshalb hier sowohl Bezüge zum Informatik-Anteil des Fachs IMP als auch zu den Lehrplaninhalten des Informatikunterrichts in der Oberstufe aufgezeigt werden.

Im Lernmaterial finden sich insbesondere Inhalte des Informatikunterrichts aus Jahrgangsstufe 9. Auf Arbeitsblatt 4 und manchen Zusatzarbeitsblättern (insbesondere derer mit offener Aufgabenstellung) ist beispielsweise die Ersetzung von Lücken in einfachen Algorithmen oder in den Zusatzblättern auch die Entwicklung einfacher Algorithmen notwendig. Damit adressiert das Lernmaterial Kompetenzen, die im Bildungsplan Informatik in Jahrgangsstufe 9 zentral sind. Hier sollen „Algorithmen

mit den Grundbausteinen Anweisung, Bedingung, Schleife und Verzweigung sowie unter Verwendung von Variablen in einer geeigneten textuellen Programmiersprache implementier[t] [werden] “(Ministerium BW, 2020, S. 17) (Ministerium BW, 2016a, S. 45). Damit das Lernmaterial auch für Schülerinnen und Schüler ohne Informatikkurs zugänglich bleibt, sind auf Arbeitsblatt 4 ausführliche Hilfe-Karten platziert, sodass die jeweiligen Aufgaben auch ohne Programmierkenntnisse bearbeitet werden können. Auch der Graph wird als Datenstruktur in der 9. Jahrgangsstufe im Informatikunterricht bereits kennengelernt, um Probleme übersichtlich darstellen zu können (Ministerium BW, 2020, S. 11; Ministerium BW, 2016a, S. 15).

Zudem fördert die Entwicklung eines Optimierungsverfahrens gezielt das algorithmische Denken. Der Bildungsplan für den IMP-Unterricht nennt explizit die vollständige Suche (Ministerium BW, 2016a, S. 58), die erfahrungsgemäß von den Lernenden gewählt wird. Dieser Ansatz soll implementiert und in Bezug auf dessen Grenzen kritisch reflektiert werden (Ministerium BW, 2016a, S. 58).

Anbindung an Curricula des Fachs Deutsch

An das Fach Deutsch gibt es ebenfalls verschiedene Anknüpfungspunkte. So lassen sich bei der Diskussion der Grenzen des entwickelten Sprachmodells gezielt Verbindungen herstellen, insbesondere im Hinblick auf grammatischen Regeln – ein zentrales Thema des Deutschunterrichts in verschiedenen Jahrgangsstufen (Ministerium BW, 2024a).

Wird das Zusatzblatt zur Untersuchung der Rolle der Daten bearbeitet, so können weitere Anknüpfungspunkte gefunden werden. Im Lernmaterial werden Wortvorschläge zweier Sprachmodelle verglichen, die mit Trainingsdatensätzen unterschiedlicher Entstehungszeit trainiert wurden. Dabei wird deutlich, dass sich die Verwendung bestimmter Wörter oder Wortsequenzen über Jahrzehnte und Jahrhunderte hinweg verändert hat. Dadurch wird das Phänomen des Sprachwandels aufgezeigt und dessen Auswirkungen auf die Wahl eines geeigneten Trainingsdatensatzes diskutiert. Während diese Aspekte bereits im Standardmaterial bei der Diskussion zu Grenzen des Modells aufgegriffen werden, erfolgt auf dem Zusatzblatt zur Rolle der Daten eine vertiefte Auseinandersetzung. Dazu passende Kompetenzen sind im Bildungsplan des Fachs Deutsch an verschiedenen Stellen verankert. So ist beispielsweise in den Jahrgangsstufen 9/10 eine Auseinandersetzung mit Anglizismen und Bedeutungsveränderungen von Wörtern als Phänomene des Sprachwandels vorgesehen (Ministerium BW, 2024a, S. 60).

14 Erfahrungswerte und didaktische Reflexion

14.1 Bisherige Erfahrungen

Um eine erste Einschätzung zur Eignung des Lehr- und Lernmaterials für mathematische Modellierungsprojekte zu gewinnen, wurde es im Rahmen des CAMMP-Projekts in verschiedenen Modellierungsveranstaltungen erprobt. Auf Basis der dabei gewonnenen Erkenntnisse wurde das Material in mehreren Iterationsschritten weiterentwickelt. Ziel dieser Testdurchläufe ist es zudem, eine vorläufige Bewertung des Interesses und der Motivation der Lernenden, der Angemessenheit des Schwierigkeitsgrades sowie des erzielten Lernfortschritts vorzunehmen. Dazu wurden die Lernenden während der Durchführung beobachtet, und es wurden nach der Veranstaltung Umfragen zur Selbsteinschätzung durchgeführt. Es sei jedoch betont, dass weiterführende Untersuchungen wie Leistungstests oder detaillierte Analysen der Antworten auf einzelne Aufgaben notwendig sind, um fundierte Aussagen über den Lernzuwachs und den langfristigen Lernerfolg treffen zu können.

Das Material wurde bereits mit rund 200 Schülerinnen und Schülern der Jahrgangsstufen 10 bis 13 des Gymnasiums eingesetzt. Tabelle 14.1 zeigt eine Übersicht über alle zum entwickelten Lehr- und Lernmaterial durchgeführten Modellierungsveranstaltungen. Es ist zu sehen, dass die Modellierungsaktivitäten sowohl in Tagesveranstaltungen als auch in Form von Unterrichtsreihen stattfanden. Auch die Teilnehmenden unterscheiden sich in den verschiedenen Durchführungen. Etwa die Hälfte der Workshops fand mit einzelnen interessierten Schülerinnen und Schülern in außerschulischen Veranstaltungen, wie etwa dem MINT EC Forum, der Modellierungs-AG oder dem Projektkurs „Mädels machen MI(N)T – mit Mathe und KI reale Probleme lösen“ (Hofmann und Schönbrodt (2024); Hofmann et al. (2024)) statt, während die weiteren Veranstaltungen mit einem Klassen- beziehungsweise Kursverband erfolgten. Somit ergeben sich sehr verschiedene Lernsettings von freiwilligen zeitintensiven Modellierungsaktivitäten an außerschulischen Lernorten bis hin zu einem unterrichtsnahen Einsatz des Materials als Unterrichtsreihe in einem Kurs- beziehungsweise Klassenverband. Die jeweilige unterschiedliche Zusammensetzung sorgt dementsprechend auch für sehr unterschiedlich heterogene Lerngruppen. Insbesondere durch den Einsatz der Materialien für Klassenverbände der Jahrgangsstufe 10, wo von einer sehr heterogenen Zusammensetzung ausgegangen werden kann, wird die Eignung des Materials für heterogene Gruppen erprobt.

Tabelle 14.1: Tabellarische Übersicht über durchgeführte Modellierungsprojekte mit dem Lernmaterial zur Wortvorhersage. In den mit * markierten Veranstaltungen wurde in einem erweiterten Zeitrahmen das Zusatzmaterial zu weiterführenden offenen Problemstellungen bearbeitet.

Veranstaltung	Anzahl Lernende	Jahrgangsstufe
Modellierungstage im Rahmen des MINT EC Forums*	30 (19 m, 11 w)	11 - 13
Unterrichtsreihe im Rahmen der Inspirationsweek der Digitalen Drehtür	6 (5 m, 1 w)	10 - 11
Unterrichtsreihe im Rahmen einer Modellierungs-AG	3 (1 m, 2 w)	10 - 13
Unterrichtsreihe mit Leistungskurs Mathematik eines Gymnasiums	13 (11 m, 2 w)	11
Modellierungstag im Rahmen des MINT Feriencamps des Cyber Forums	5 (4 m, 1 w)	10 - 11
Modellierungstag mit Schulklasse eines Gymnasiums	18 (6 m, 12 w)	10
Unterrichtsreihe im Rahmen des Schnupperkurses am KIT*	8 (6 m, 2 w)	9 - 12
Modellierungstag mit Schulklasse eines Gymnasiums	23 (10 m, 12 w, 1 d)	10
Modellierungstag mit Leistungskurs Mathematik eines Gymnasiums	17 (13 m, 4 w)	12
Unterrichtsreihe im Rahmen des Projektkurses „Mädels machen MI(N)T“	10 (10 w)	10 - 12
Modellierungstag mit Schulklasse eines Gymnasiums	23 (14 m, 9 w)	10
Modellierungs-Doppelstunde mit Schulklasse eines Gymnasiums	18 (10 m, 8 w)	10
Modellierungs-Doppelstunde mit Schulklasse eines Gymnasiums	19 (11 m, 8 w)	10
Unterrichtsreihe im Rahmen des Projektkurses KI für Mädels	6 (6 w)	10 - 13

Beobachtungen und daraus resultierende Anpassungen des Lernmaterials

Es folgt eine Übersicht über zentrale Beobachtungen aus den Durchführungen der Modellierungsveranstaltungen. Dabei wird auf die Antworten der Schülerinnen und Schüler während der Plenumsphasen eingegangen, und die beobachteten Lernschwierigkeiten werden beschrieben. Gleichzeitig wird dargelegt, an welchen Stellen im Material Änderungen vorgenommen wurden, um den beobachteten Lernschwierigkeiten zu begegnen.

In der ersten Version dieses Lernmaterials wurden deutlich weniger Schritte im Hintergrundcode ausgelagert, und die Implementierung des Uni-, Bi- und Tri-Gramm-Modells musste unter Anleitung selbst vorgenommen werden. Aufgrund der begrenzten Programmierkenntnisse der Lernenden erwies sich dieser Ansatz als zu anspruchsvoll und technisch fokussiert, wodurch die mathematischen Erkenntnisse in den Hintergrund traten. Die Automatisierung des Prozesses der Berechnung und Abspeicherung der relativen Häufigkeiten wurde daher größtenteils ausgelagert. Die ursprünglichen Aufgaben wurden zu Zusatzblättern umgestaltet. Diese können von Schülerinnen und Schülern, die stärker an der technischen Umsetzung interessiert sind, bearbeitet werden.

Bei der Bearbeitung des ersten Arbeitsblattes konnten ansonsten kaum Schwierigkeiten festgestellt werden. Als eine Hürde in den ersten Durchführungen stellte sich jedoch das Aufstellen einer allgemeinen Formel für die relative Häufigkeit heraus. Viele forderten an dieser Stelle die Unterstützung der Betreuenden ein. Es konnte beobachtet werden, dass ihnen die Verallgemeinerung vom Beispiel schwer viel. Daher wurde ein Zwischenschritt hinzugefügt, indem die Berechnung der relativen Häufigkeit erst einmal für ein konkretes Beispiel formal notiert werden soll. Zudem wurde der Schritt des Aufstellens einer allgemeinen Formel zweigeteilt: Zunächst sollen diejenigen Größen genannt werden, welche für die Berechnung der relativen Häufigkeit notwendig sind. Anschließend soll die Formel aufgestellt werden.

In der nachfolgenden Besprechungsphase wurden nach der Sicherung auch Probleme des Modells und Ideen für mögliche Modellverbesserungen gesammelt. Häufig wurde hier als Erstes die Verwendung eines größeren Trainingsdatensatzes genannt. Die entscheidende Rolle der Daten für Sprachmodelle scheint somit deutlich geworden zu sein. Ebenfalls wurde in nahezu jeder Durchführung vorgeschlagen, dass die Wortvorgeschichte für einen stärkeren Kontextbezug der Vorschläge erweitert werden kann und dass bei einer ungesesehenen Wortvorgeschichte die im Trainingsdatensatz am häufigsten enthaltenen Wörter vorgeschlagen werden. Damit wurde bereits die zugrunde liegenden Ideen sowohl des Tri-Gramm-Modells als auch des Uni-Gramm-Modells genannt.

Bei der Bearbeitung des zweiten Arbeitsblattes ist aufgefallen, dass die Anpassung des Übergangsgraphen an den erweiterten Trainingsdatensatz äußerst selten Probleme bereitete. Es konnte allerdings beobachtet werden, dass Lernende Schwierigkeiten bei der Erstellung des Übergangsgraphen für das Tri-Gramm-Modell hatten. Viele hatten Probleme, zu erkennen, dass hier zwei Wörter zu einem Knoten zusammenge-

fasst werden müssen, und teilweise fiel auch die Berechnung der relativen Häufigkeiten schwer. Es wurden daher zwei ausführliche aufeinander aufbauende Tippkarten ergänzt. Auf diesen wird das Erstellen des Übergangsgraphen im Tri-Gramm-Modell am Beispiel eines Übergangs Schritt für Schritt erklärt.

Weitere Anpassungen wurden in Bezug auf die Kombination der N -Gramm-Modelle vorgenommen. Die Rückfallstrategie war zu Beginn kein Teil des Lernmaterials. Bei den ersten Durchführungen konnte allerdings beobachtet werden, dass diese Strategie im Gegensatz zur Interpolation gehäuft als Kombinationsmöglichkeit vorgeschlagen wurde. Zum Beispiel äußerte eine Teilnehmerin die Idee „Können wir nicht einfach eine kürzere Wortsequenz suchen, wenn wir die lange [Wortsequenz] im Datensatz nicht finden?“¹. Zur Rückfallstrategie wurde deshalb ein Zusatzblatt entwickelt, sodass interessierte Lernende ihre vorgeschlagene Idee auch tatsächlich weiterverfolgen können. Bei der Bearbeitung von Arbeitsblatt 3 konnten keine Hürden festgestellt werden.

Bei der Sammlung verschiedener Bewertungskriterien wurden die Zeitersparnis und die Einsparung der Tastenanschläge besonders häufig genannt. Die Liste der möglichen Bewertungskriterien wurde daher angepasst und um diese beiden Vorschläge ergänzt. Zudem wurde die Trefferrate als mögliches Bewertungskriterium benannt. Zum Beispiel äußerte eine Teilnehmerin „Man könnte das gewünschte Wort mit dem vorgeschlagenen Wort vergleichen und dann alle [Übereinstimmungen] zählen.“. In einer einzelnen Durchführung wurde außerdem angeregt, die Übergangswahrscheinlichkeiten für den Übergang von verschiedenen Wortvorgeschichten zu dem Wort, das man sich jeweils als Vorschlag wünscht, zu erfassen – ein Ansatz, der letztlich der Kreuzentropie zugrunde liegt. Dass dieser Vorschlag sonst nicht geäußert wurde, ist angesichts der anderen intuitiven Kriterien nicht überraschend.

Bei der Bearbeitung von Arbeitsblatt 4 konnten die meisten Schwierigkeiten beobachtet werden, was aufgrund der hohen Komplexität des Arbeitsblattes zu erwarten war. Zu den Herausforderungen zählte zum einen die Bestimmung der Modellwahrscheinlichkeit. Viele Lernende hatten zwar die Idee, ein Baumdiagramm zu zeichnen, um die Berechnung der Modellwahrscheinlichkeit zu erleichtern, doch der Umgang mit einer großen Anzahl an Pfaden fiel ihnen schwer. Daher wurde ein zusätzlicher Tipp hinzugefügt, der zeigt, wie nicht relevante Pfade durch einen einzigen zusammengefasst angedeutet werden können.

Zum anderen wurde festgestellt, dass die Grenzwertuntersuchung der Logarithmusfunktion schwer fiel. Diese soll bei der Bewertung des nicht geglätteten Sprachmodells vorgenommen werden. Besonders Schülerinnen und Schüler der 10. Jahrgangsstufe hatten Schwierigkeiten, vermutlich da sie weder die Logarithmusfunktion noch deren Grenzverhalten im Unterricht behandelt hatten (vgl. Abschnitt 13.3). Die Kurzeinführung zur Logarithmusfunktion im Plenum im Rahmen der Einführung der Kreuzentropie schien daher dafür, dass viele Inhalte neu waren, zu knapp gewesen zu sein. Es wurde daher in den folgenden Durchführungen für Lerngruppen der 10. Jahrgangsstufe ausführlich über den Graphen der Logarithmusfunktion

¹Rechtschreib- und Interpunktionsfehler wurden in diesem und den nachfolgenden Schülerzitaten zur besseren Lesbarkeit korrigiert.

und deren Eigenschaften im Plenum gesprochen sowie bereits dessen Grenzverhalten (propädeutisch) diskutiert. Zusätzlich wurden weitere gestufte Tipps ergänzt, die schrittweise weitere Informationen oder gezielte Leitfragen zur Lösung der Aufgabe bereitstellen.

Auch die Termumformungen zur Berechnung der Kreuzentropie für das Zufalls-Modell bereiteten einigen Lernenden Schwierigkeiten. Daher wurden auch hier ausführliche gestufte Tippkarten ergänzt sowie eine Informationskarte zu den Logarithmusgesetzen hinzugefügt.

In der anschließenden Besprechung sollte erklärt werden, weshalb die Modellwahrscheinlichkeit sehr klein ist und welche Vorteile das Logarithmieren der Modellwahrscheinlichkeit hat. In allen Durchführungen konnten die Lernenden dazu korrekte Erklärungen formulieren. Darunter zählt zum Beispiel die folgende Erklärung:

„Die Werte werden so klein, denn wenn man $1/6$ mal $1/6$ mal $1/6$ und so weiter rechnet ist das Ergebnis fast null. Wenn man aber den Logarithmus nimmt, berechnet man $1/6$ plus $1/6$ plus $1/6$ und so weiter und das ist für den Computer leichter zu rechnen.“

Außerdem deuten die Wortmeldungen der Teilnehmenden darauf hin, dass sie bereits durch unsystematisches Ausprobieren verschiedener Gewichtungen und das Bestimmen der Kreuzentropie erkannt haben, dass eine gezielte Suche nach dem Minimum der Kreuzentropie eine sinnvolle Strategie zur Festlegung der optimalen Gewichte ist.

In der kurzen Variante des fünften Arbeitsblattes fielen keine Schwierigkeiten auf. Die längere Version des Arbeitsblattes wurde nur mit leistungsstarken und besonders motivierten Lerngruppen durchgeführt (vgl. Tabelle 14.1). Dabei konnten alle Zweier-Teams mit gar keiner bis wenig Unterstützung einen Optimierungsalgorithmus entwickeln. Abbildung 14.1 zeigt beispielhaft eines von einer Gruppe entwickelte Optimierungsverfahren. Die vollständige Suche war dasjenige Optimierungsverfahren, das als einziges von den Lernenden entwickelt wurde und zwar nach ihrer Aussage in vielen Fällen auch ohne dass davor die Tipps, die ohnehin zu diesem Optimierungsverfahren führen, genutzt wurden. Zudem schien die Problemstellung die Lernenden zu aktivieren und ihren Ehrgeiz zu wecken, unter allen Teams die genaueste Lösung zu finden.

In der abschließenden Diskussion über Probleme und Grenzen des Modells beteiligten sich die Teilnehmenden in den meisten Durchführungen intensiv. Wiederholt genannte Kritikpunkte am entwickelten Sprachmodell waren:

- die fehlende Berücksichtigung grammatischer Regeln,
- die fehlende Anpassung der Wortvorschläge an den persönlichen Schreibstil,
- teilweise Rechtschreibfehler in den Wortvorschlägen, da kein Abgleich mit einem Wörterbuch erfolgt,
- die nur begrenzte Erfassung des Satzkontextes.

```

# Hier ist Platz für deinen Code!
min = 10.0
gz = (0,0,0,0)

for g1 in 0:100
    for g2 in 0:(100-g1)
        for g3 in 0:(100-g1-g2)
            g0 = 100 - g1 - g2 - g3

            tmp = KE(g0/100,g1/100,g2/100,g3/100)
            if tmp < min
                min = tmp
                gz = (g0,g1,g2,g3)
            end
        end
    end
print(min)
print(gz)

```

Abbildung 14.1: Entwickeltes Optimierungsverfahren einer Lerngruppe

Auch bei der Sammlung von problematischen Aspekten bei der Nutzung und Entwicklung von Sprachmodellen brachten sich die meisten sehr aktiv ein. Unter den bereits in Abschnitt 13.2.6 genannten problematischen Punkten, wurde besonders häufig die Beeinflussung des Sprachverhaltens genannt. So wurde beispielsweise diskutiert, dass Texte durch die Nutzung solcher Systeme anders klingen als man sie selbst formulieren würde, und dass es dadurch leichter fällt, sich als jemand anderes auszugeben. Zudem wurden Bedenken hinsichtlich eines möglichen Verlusts von Kreativität und Rechtschreibkompetenzen geäußert, wenn man sich zu stark auf Assistenzsysteme beim Schreiben verlässt. In einer Durchführung wurde außerdem kritisch hinterfragt, welche Rolle Vorurteile in den Trainingsdaten spielen und wie sich diese auf die generierten Texte auswirken können.

Selbsteinschätzung der Lernenden laut der schriftlichen Befragung

Ergänzend zu den Beobachtungen in den Durchführungen werden hier die Ergebnisse der schriftlichen Befragung zusammengefasst. Eine Übersicht der Evaluationsergebnisse ist in Anhang C.2 zu finden. Es ist anzumerken, dass nicht alle Lernenden an der Umfrage teilgenommen haben und auch diejenigen, die teilnahmen, nicht unbedingt alle Fragen beantwortet haben.

In der Evaluation stimmten 120 von 141 Lernenden der Aussage, dass sie das mathematische Modellieren durch den Workshop besser begriffen haben, eher oder völlig zu, und auch beim Lernzuwachs wurde die mathematische Modellierung in vielen Fällen hervorgehoben. Dies deutet darauf hin, dass der Workshop einen Beitrag dazu leistet, das Konzept der mathematischen Modellierung besser zu verstehen. Die folgenden beispielhaften Antworten auf die Frage “Was hast du für dich persönlich durch die Teilnahme am Workshop gelernt?” unterstreichen zudem, dass die Bedeu-

tung der Mathematik für den Alltag erfahrbar geworden ist, was eines der Teilziele der entwickelten Lernmaterialien darstellt (vgl. Abschnitt 4.1):

- „Ich habe gelernt, dass Modellierung im Alltag Anwendung findet“
- „Ich fand den Einblick in die Generierung von Wortvorschlägen sehr interessant und es war ein sehr gutes Beispiel dafür, wie mathematische Modellierung im Alltag Anwendung findet.“
- „Wo im Alltag Mathe und Informatik dahinter steckt“
- „Wie manche Dinge so stark von der Mathematik abhängen“
- „Ich habe gelernt, wie man Stochastik bei realen Problemen anwendet.“
- „Wie Mathe in der Realität benutzt wird“

Auch die Gestaltung und der Aufbau der Arbeitsblätter wurden häufig gelobt, wie die folgenden beispielhaften Zitate zeigen:

- „Die Präsentation und die Arbeitsblätter waren gut gestaltet und uns wurde alles gut erklärt.“
- „Vielen Dank für diesen Workshop, er ist sehr hilfreich und interessant, und wie ihr die Themen erklärt und wie die Arbeitsblätter aufgebaut sind, ist sehr toll.“
- „die gute Strukturierung des Erarbeiten der neuen Inhalte“

Zudem stimmten 120 von 140 Lernenden der Aussage „Die Gestaltung der Arbeitsblätter fand ich gut“ völlig oder eher zu, und die allermeisten Lernenden (133 von 143) fanden, dass die Inhalte klar vermittelt wurden.

Sowohl die automatischen Rückmeldungen als auch die Tipps wurden insgesamt als hilfreich empfunden. Der Aussage, dass sie hilfreich seien, stimmten im Fall der Tipps 91 von 137 und im Fall der Rückmeldungen 110 von 140 Teilnehmende völlig oder eher zu.

Der Schwierigkeitsgrad der Aufgaben scheint insgesamt angemessen gewesen zu sein. Die Mehrheit der Lernende (110 von 140) stimmte der Aussage, dass die Aufgaben zu einfach seien, gar nicht oder eher nicht zu. Ähnlich sehen die Antworten auf die Aussage, dass die Aufgaben zu schwierig sind, aus. Nur sechs Personen stimmten der Aussage vollkommen zu, 44 Lernende stimmten dem eher, 66 stimmten eher nicht und 27 gar nicht zu. Zusätzlich unterstrichen wird diese Feststellung von verschiedenen Freitextantworten, darunter zum Beispiel die Antwort „Verständliche Texte undfordernde, aber nicht unmögliche Aufgaben, haben das Verständnis des Themas sehr erleichtert“. Die Mehrheit (95 von 141) stimmten außerdem vollkommen oder eher zu, dass sie die Aufgaben eigenständig bearbeiten konnten.

Der Umgang mit der Programmiersprache stellte für mehr als die Hälfte der Teilnehmenden keine Herausforderung dar. Dies mag daran liegen, dass außerhalb der Zusatzblätter nur einzelne Lücken im Code ergänzt werden müssen und benötigte Befehle in der Aufgabenstellung angegeben sind. 48 von 138 Lernenden stimmten

der Aussage, dass der Umgang mit der Programmiersprache ihnen schwer fiel, allerdings eher oder völlig zu. Wie bereits in Abschnitt 14.1 beschrieben, wurde daher der technisch fokussierte Anteil insbesondere in den ersten beiden Arbeitsblättern erheblich reduziert.

Die Plenums- und Arbeitsphasen sowie die Pausen scheinen zeitlich geeignet platziert worden zu sein. 73 von 143 Personen stimmten der Aussage vollkommen zu, dass die Lern- und Arbeitszeit angemessen war, und 50 stimmten ihr eher zu.

Das selbstgesteuerte Arbeiten an den Arbeitsblättern wurde in der schriftlichen Evaluation als sehr positiv hervorgehoben. So wurde auf die Frage „Was hat dir besonders am Workshop gefallen?“ unter anderem die Antworten

- „Freies und selbstständiges Arbeiten“
- „dass wir mehr oder weniger alleine und selbstständig arbeiten durften“
- „Selbstständigkeit, aber wenn man eine Frage hatte waren die Betreuer hilfsbereit“
- „Die Selbstständigkeit, aber auch, wie der Workshop generell aufgebaut war“

gegeben.

In den Freitextantworten wurde zudem auch die Aktualität (1x) und Relevanz des Themas (1x) sowie der Alltagsbezug des Themas (6x) positiv hervorgehoben.

Eines der Hauptziele des Workshops ist die Aufklärung über die Funktionsweise von KI-Systemen (vgl. Abschnitt 4.1). Einige der Freitextantworten legen nahe, dass mit diesem Workshop dieses Ziel erreicht werden konnte. Dazu zählt die Antwort zur Frage, ob etwas besonders gut gefallen hat und wenn ja, was: „Ja, zu sehen, dass aus kleinen Schritten ein Programm zusammengesetzt wurde, das einen in sich stimmigen Satz ausgeben konnte.“ und die Antworten zum Lernzuwachs:

- „viel über KI-Wortvorschläge und wie man diese programmiert“
- „wie KIs für Wortvorschläge funktionieren und auf was für Dinge diese achten.“
- „Wie KI überhaupt funktioniert!“
- „[...] wie KI's für Wortvorschläge funktionieren und auf was für Dinge diese achten.“

Die Bewertung des Workshops insgesamt fiel sehr gut aus. 128 von 153 Teilnehmenden stimmten der Aussage, dass sie den Workshop weiterempfehlen würden, eher oder völlig zu. Zudem gaben 71% der Lernenden dem Workshop die Schulnote sehr gut und alle anderen bewerteten den Workshop mit der Note gut.

Die Evaluation des Lernmaterials im Rahmen des CAMMP-Projekts zeigt damit insgesamt eine positive Rückmeldung der Lernenden. Sowohl die Gestaltung und der Aufbau der Arbeitsblätter als auch die Betreuung und die Möglichkeit zum selbstgesteuerten Arbeiten wurden positiv bewertet, und auch die Schwierigkeit der Aufgaben scheint angemessen zu sein. Obwohl weitere detaillierte Untersuchungen notwendig sind, um den langfristigen Lernerfolg umfassend zu beurteilen, deuten die

vorliegenden Ergebnisse darauf hin, dass das entwickelte Lernmaterial die Vermittlung von KI-Kompetenzen und mathematischen Inhalten im Kontext der Wortvorhersage sinnvoll ermöglichen.

14.2 Reflexion

Dieses Kapitel reflektiert, inwiefern die in Kapitel 7 genannten Design-Prinzipien in diesem Lehr- und Lernmaterial umgesetzt werden.

Die Entwicklung eines Sprachmodells erfolgt in der Praxis aufgrund der interdisziplinären Natur der Problemstellung automatisch in Zusammenarbeit mit unterschiedlichen Fachrichtungen (vgl. Kapitel 7, DP3). Neben mathematischen und informatischen Kompetenzen ist bei der Entwicklung außerdem Wissen aus der Linguistik notwendig. Dieser Interdisziplinäre Charakter spiegelt sich auch im Lehr- und Lernmaterial wieder. Der Fokus liegt auf dem Fach Mathematik, es gibt aber Anknüpfungspunkte zu anderen Fächern (vgl. Abschnitt 13.3) wie beispielsweise zum Fach Deutsch, indem thematisiert wird, welche syntaktischen Fehler beim durch das Sprachmodell generierten Text auftreten.

Auch innermathematisch ermöglicht die Problemstellung Verknüpfungen (vgl. Kapitel 7, DP3). So liegt bei dieser Problemstellung der Fokus zwar auf stochastischen Konzepten, dennoch werden aus dem Bereich Analysis Eigenschaften der Logarithmusfunktion für die Bewertung des *N*-Gramm-Modells näher untersucht, sowie ein Optimierungsproblem für das Festlegen der Gewichte aufgestellt (vgl. Abschnitt 13.3).

Assistenzsysteme, die Smartphonenuutzende beim Tippen von Nachrichten unterstützen, stellen ein Kontext aus der Lebenswelt der Lernenden dar. Diese Systeme verwenden sie häufig mehrmals täglich, wenn sie mit Freunden chatten. Der Problemkontext wird daher als relevant für die Lernenden eingeschätzt (vgl. Kapitel 7, DP5). Schülerinnen und Schüler kennen und verstehen somit die praktische Situation und können dementsprechend das Modell eher bewerten, wie wenn sie die Erfordernisse des Modells nicht nachvollziehen können.

Die Frage nach der Funktionsweise von Assistenzsystemen wie dem System zur Wortvorhersage lässt sich zudem als authentisch einstufen, da sie bereits von Forschenden bei der Entwicklung solcher Systeme untersucht wurde (vgl. Kapitel 7, DP5). Im vorliegenden Lernmaterial wird das *N*-Gramm-Modell als Sprachmodell verwendet, ein im Bereich NLP etabliertes Verfahren (Jurafsky & Martin, 2023). Bis vor wenigen Jahren wurden *N*-Gramm-Modelle auch in namhaften Assistenzsystemen wie Google Keyboard eingesetzt (Hard et al., 2019, S. 1). Damit sind die zugrunde liegenden mathematischen Methoden und Verfahren als authentisch anzusehen (vgl. Kapitel 7, DP5). Seit 2019 verwendet Google jedoch neuronale Netze, die auch übergreifendere Kontextbezüge erfassen können (Hard et al., 2019, S. 2). Neuronale Netze und insbesondere solche mit einer sogenannten Transformer-Architektur, die speziell für die Verarbeitung sequenzieller Daten entwickelt wurde, haben sich inzwischen als

Grundbaustein moderner Sprachmodelle, wie etwa bei ChatGPT, etabliert (Vaswani et al., 2017, S. 3 - 5).

Dennoch bietet es sich aus mathematikdidaktischer Perspektive an, *N*-Gramm-Modelle im Unterricht zu thematisieren. Im Vergleich zu neuronalen Netzen ermöglichen *N*-Gramm-Modelle einen leichteren Einstieg, um zentrale mathematische Konzepte von ML-Methoden zu vermitteln (vgl. Kapitel 7, DP4). Das grundlegende Konzept – die Schätzung der Übergangswahrscheinlichkeiten über relative Häufigkeiten der Wortsequenzen – ist für die Lernenden intuitiv verständlich. *N*-Gramm-Modelle erlauben es zudem, grundlegende Schritte der Entwicklung eines Sprachmodells nachzuvollziehen und insbesondere das Extrahieren von Informationen aus Texten als Schlüsselprozess zu verstehen. Am Beispiel des *N*-Gramm-Modells lassen sich insgesamt alle zentralen Kompetenzen fördern, welche die Funktionsweise von KI-Systemen betreffen und in Kapitel 6 als übergreifende Lernziele formuliert sind (vgl. Abschnitt 13.1).

Aus mathematischer Perspektive eignet sich das *N*-Gramm-Modell, weil für dessen Verständnis auf einem Grundlevel lediglich relative Häufigkeiten und mehrstufige Zufallsexperimente notwendig sind. So kann an schulisches Vorwissen angeknüpft und auf bereits bekannten schulmathematischen Inhalten aufgebaut werden. Gleichzeitig können Grundprinzipien von Konzepten grundlegender Inhalte der Hochschulstochastik wie Markov-Ketten und die Maximum-Likelihood-Schätzung problembezogen veranschaulicht werden. Diese werden im Lernmaterial zwar nicht explizit benannt, können aber möglicherweise dazu beitragen, dass Lernenden die Inhalte in ihrem Studium wiedererkennen und direkt ein Anwendungsbeispiel kennen.

Das *N*-Gramm-Modell bietet zudem die Möglichkeit, ein erstes Modell mithilfe von Modifikationen Schritt für Schritt zu verbessern. Um die kognitive Belastung zu reduzieren, wird daher zunächst das einfachste *N*-Gramm-Modell – das Bi-Gramm-Modell – eingeführt (vgl. Kapitel 7, DP4). Anhand eines kleinen Beispieldatensatzes ermitteln die Lernenden relative Häufigkeiten der im Datensatz vorkommenden Wortsequenzen. Die Automatisierung dieses Prozesses wird anschließend in der digitalen Lernumgebung an das digitale Werkzeug ausgelagert. Durch eine Sammlung von Problemen des bis dahin entwickelten Modells und Modellverbesserungen wie die Kombination verschiedener *N*-Gramm-Modelle mit unterschiedlich langer Worthistorie kann das Modell Schritt für Schritt präzisiert und ein tatsächlich einsatzfähiges Sprachmodell konstruiert werden, ohne kognitiv zu überfordern.

Der Modellierungskreislauf wird dabei mehrfach vollständig durchlaufen, auch wenn die systematische Bewertung erst am Ende erfolgt. Die konkrete Modellverbesserung wird durch die Lernumgebung vorgegeben, jedoch werden im Vorfeld zuerst die Probleme des bisher entwickelten Modells sowie Ideen zur Verbesserung im Plenum gesammelt, um die Überlegungen der Lernenden einzubeziehen. Alternativ kann die Rückfallstrategie als zusätzliche Modellverbesserung auf einem optionalen Arbeitsblatt erkundet werden.

Die Reflexion über die Probleme des *N*-Gramm-Modells sowie eine Rückbesinnung auf die zugrunde liegenden Modellannahmen verdeutlichen, dass die Modellierung der realen Situation – der Generierung von Sprache – nicht beliebig genau sein kann.

Die Komplexität der Sprachproduktion macht es schwierig, diese vollständig in einem Modell abzubilden und zeigt so gleichzeitig die Grenzen von KI-Systemen auf (vgl. Kapitel 7, DP4).

Zusätzlich wird in einem Zusatzblatt mit Schülerinnen und Schülern erarbeitet, wie das entwickelte *N*-Gramm-Modell nicht nur zur Wortvorhersage, sondern auch zur Generierung von ganzen Texten genutzt werden kann. Dadurch wird gezeigt, dass Modelle mit kleinen Anpassungen häufig für unterschiedliche Zwecke eingesetzt werden können. Diese Tatsache bewusst zu machen, zählt laut Pohjolainen und Heiliö (2016, S. 4) zu den Aspekten, die gutes Lernmaterial zur Modellierung zeigen sollte.

Neben den in Kapitel 9 genannten übergreifenden methodischen Maßnahmen, die die Kommunikationsfähigkeit fördern sollen, werden im Lernmaterial Aufgaben integriert, die ebenso dazu beitragen sollen. Dazu zählen Aufgaben zur Sammlung von Problemen und Modellverbesserungen sowie Reflexionsfragen, die im Rahmen der Bewertung der Modelle beantwortet werden sollen. Des Weiteren erfordern besonders die weiterführenden offenen Aufgaben einen Austausch zwischen den Lernenden, um gemeinsam passende Algorithmen zu entwickeln (vgl. Kapitel 7, DP2).

In diesem Lernarrangement wird eine digitale Lernumgebung eingesetzt, um eine bestmögliche Unterstützung bei der Bearbeitung des Modellierungsproblems zu gewährleisten (vgl. Kapitel 7, DP1). Der Einsatz von didaktisch aufgearbeiteten Jupyter Notebooks als Lernumgebung erleichtert die Verarbeitung umfangreicher Daten erheblich, und zwar sowohl in der Trainings- als auch in der Testphase. In der Trainingsphase kann das *N*-Gramm-Modell anstatt wie anfangs händisch mit drei Beispielsätzen nun mit größeren Datensätzen trainiert werden. Ebenso unterstützt das digitale Tool die umfassende Bewertung der Modellergebnisse, indem es die Berechnung der Modellwahrscheinlichkeit auf einem Testdatensatz mit über zweitausend Sätzen problemlos ermöglicht. Zudem können in der digitalen Lernumgebung die Kreuzentropie von interpolierten Modellen mit unterschiedlicher Gewichtung durchgeführt und damit eine Variation der Parameter des *N*-Gramm-Modells stattfinden. Die digitale Lernumgebung ermöglicht es auch, die Durchführung einzelner Verfahren, wie beispielsweise das Optimierungsverfahren auf Arbeitsblatt 5 an das digitale Werkzeug auszulagern. Eine implementierte Anwendung, die basierend auf dem trainierten Modell Wortvorschläge zu eingegebenen Sequenzen generiert, ermöglicht zusätzlich die Simulation eines realen Anwendungsfalls. Dadurch können die Funktionsweise des Systems direkt reflektiert und dessen Grenzen erkundet werden. Außerdem wird eine Verknüpfung der Entwickler- und Anwenderperspektive sowie der theoretischen Überlegungen und Praxis erzielt, wodurch die praktische Nutzen mathematischer Methoden einmal mehr deutlich gemacht wird. Zugleich erhofft man sich dadurch eine Steigerung der intrinsischen Motivation, indem Lernende direkte Erfolge ihrer Überlegungen sehen.

Insgesamt bietet die Unterrichtseinheit die Möglichkeit, Lernenden einen Einblick in die Verarbeitung natürlicher Sprache zu geben und damit die Funktionsweise der Mathematik moderner Anwendungen wie generativen Sprachmodellen verständlich zu machen. Zugleich ermöglicht sie das angeleitete Modellieren eines authentischen, mathematisch reichhaltigen Problems unter Anwendung von Inhalten aus dem Bildungsplan.

15 Ausblick

Dieser Abschnitt gibt einen Ausblick auf mögliche Weiterentwicklungen des Lehr- und Lernmaterials zur Wortvorhersage. Zunächst soll der Fokus auf weiterführende Forschungsaspekte im methodisch-didaktischen Bereich gelegt werden.

Eine perspektivisch denkbare Weiterentwicklung betrifft die Gestaltung der digitalen Lernumgebung. Derzeit basieren die Jupyter Notebooks des Lernmaterials auf der Programmiersprache Julia. Eine Einschränkung dieser Implementierung ist, dass interaktive Widgets wie beispielsweise Freitextantworten nicht direkt im Notebook integriert werden können. Daher notieren die Lernenden ihre Freitextantworten derzeit auf einem separaten Antwortblatt. Das hat zur Folge, dass die Antworten nicht alle gesammelt vorliegen und zum Beispiel berechnete Lösungen getrennt von deren Interpretation stehen. Eine mögliche Alternative wäre die Umsetzung des Materials in Python, wodurch eine vollständig digitale Bearbeitung innerhalb der Lernumgebung möglich wäre. Zudem könnten weitere interaktive Elemente hinzugefügt werden. Eine Auswahl ist beispielhaft im Lernmaterial zur Spracherkennung zu sehen (vgl. Anhang B).

Aktuell wird der Übergang von der Datenanalyse zur beurteilenden Statistik auf Arbeitsblatt 1 stärker angeleitet. Der Hauptgrund liegt darin, dass viele mathematische Inhalte in kurzer Zeit vermittelt werden müssen, um alle für ML-Methoden als relevant erachteten Lernziele adressieren zu können. Will man stärker auf einen Aufbau von Kompetenzen in der Stochastik fokussieren, so ist hier mit einem anderen Zeitrahmen ein selbstständigeres Erarbeiten dieses Übergangs möglich. Eichler und Vogel (2009, S. 159 - 164) schlagen am Beispiel einer Aufgabe zum Schätzen des Anteils roter Schokolinsen einer Süßigkeitenpackung vor, Lernende den Übergang von der Datenanalyse zur beurteilenden Statistik selbstständig und explorativ erarbeiten zu lassen. Anhand der Analyse mehrerer Schokolinsen-Packungen soll die Wahrscheinlichkeit, mit der eine Schokolinsen rot ist, geschätzt und eine Prognose über die von der Produktion vorgesehene Häufigkeit der Farbe erstellt werden. Leitend bei diesem Unterrichtsvorschlag ist die Frage „Gilt das immer?“, wodurch die Datenanalyse und die Wahrscheinlichkeitsrechnung miteinander verbunden werden. Angelehnt an dieses Vorgehen können Lernenden nach der Analyse der Trainingsdaten die Fragen gestellt werden: Gelten die im kleinen Trainingsdatensatz beobachteten relativen Häufigkeiten immer? Wie sinnvoll ist es, für das Aufstellen von Prognosen einen Trainingsdatensatz zu wählen, der nur aus drei Sätzen besteht? Gelten die in einem größeren Trainingsdatensatz beobachteten relativen Häufigkeiten immer? Für welche Art von Texten können die relativen Häufigkeiten besonders gute Prognosen sein und für welche eher nicht?

Aus mathematisch-inhaltlicher Sicht ergeben sich ebenfalls interessante anknüpfende oder vertiefende Ansätze, die in das entwickelte Modell integriert werden könnten. Bei der Entwicklung des Lernmaterials wurde statt des mathematisch formalen Zugangs über die Bestimmung eines Schätzers für die Übergangswahrscheinlichkeiten die intuitive Interpretation der Schätzung/ Näherung der Übergangswahrscheinlichkeiten mithilfe der relativen Häufigkeiten gewählt. Mit leistungsstarken Schülerinnen und Schülern der Kursstufe und entsprechendem Zeitrahmen wäre es jedoch denkbar, die Bestimmung eines Punktschätzers für das Uni-Gramm-Modell anhand des Maximum-Likelihood-Kriteriums — wie in Abschnitt 11.2 erläutert — zu erarbeiten. Die Punktschätzung wird in der Schule nur sehr oberflächlich ohne das Einführen des Konzeptes einer Schätzfunktion behandelt (Biehler et al., 2023, S. 249). Die Multinomialverteilung kann als Verallgemeinerung der aus dem Mathematikunterricht bekannten Binomialverteilung eingeführt werden. Die Gültigkeit der Logarithmusungleichung $\log x \leq x - 1$ für $x > 0$ kann zum Beispiel durch eine graphische Visualisierung der Funktionen $f(x) = \log x$ und $g(x) = x - 1$ ersichtlich gemacht werden. Alle weiteren Abschätzungen sind elementarer Natur und mit dem Vorwissen aus der Schule verständlich. In diesem Zusammenhang könnte auch die Auswirkung der Stichprobengröße, also der Größe des Trainingsdatensatzes, auf die Genauigkeit der Schätzung für die Wahrscheinlichkeit im Uni-Gramm-Modell untersucht werden. Das kann entweder formal wie in Abschnitt 11.2.1 untersucht werden oder eher denkbar ist eine Untersuchung der Streuung der relativen Häufigkeiten für unterschiedliche Datensatzgrößen.

Auf einem fortgeschritteneren Niveau könnten außerdem weitere Optimierungsverfahren zur Lösung des in Aufgabenblatt 5 formulierten Optimierungsproblems untersucht werden. Auch das Erarbeiten weiterer Verfahren zur Glättung des Sprachmodells und deren Vergleich ist denkbar.

Teil IV

Lehr- und Lernmaterial zur Spracherkennung

16 Einleitung

Die Spracherkennung spielt eine entscheidende Rolle in zahlreichen Anwendungen des täglichen Lebens sowie in der Industrie und Forschung. Sie bezeichnet den Prozess des maschinellen Erkennens von gesprochener Sprache und somit des Umwandelns mündlicher Kommunikation in schriftliche Textform. Für den Menschen ist das Verstehen von Sprache intuitiv, unterschiedliche Sprecher, Hintergrundgeräusche oder verschiedene Aussprechvarianten bereiten in der Regel keine Probleme. Die Annahme, automatische Spracherkennung sei einfach, erscheint daher naheliegend, trifft allerdings definitiv nicht zu. Die Variabilität der gesprochenen Äußerung bereitet bei der Klassifizierung der Äußerung große Schwierigkeiten. Wie dieses Problem mithilfe von Mathematik gelöst werden kann, wird in den folgenden Kapiteln erläutert.

Bereits in den sechziger Jahren wurden erste sehenswerte Ergebnisse in der Erkennung kontinuierlich gesprochener Sprache erzielt (Pfister & Kaufmann, 2017, S. 325). In den siebziger Jahren wurden die Systeme unter anderem durch die Entwicklung der *dynamischen Zeitanpassung* noch leistungsfähiger (vgl. Abschnitt 17.3). Vor allem seit der Ära der Künstlichen Intelligenz hat Spracherkennung an Bedeutung gewonnen, da sie insbesondere bei der natürlichsprachlichen Interaktion mit Assistenzsystemen und humanoiden Robotern Anwendung findet. Das Spracherkennungssystem Siri des Herstellers Apple kam 2011 auf den Markt und war damit nicht das erste leistungsfähige Assistenzsystem, das fließende Sprache verarbeitete, aber das erste, welches einer breiten Öffentlichkeit zugänglich gemacht wurde (Rauch, 2016). Heute sind solche Sprachassistenten, die entweder direkt am Smartphone genutzt werden oder als kleine Computer mit Lautsprecher in vielen Haushalten anzutreffen sind, weit verbreitet. Vor allem die Bedienung durch Sprache machen diese Assistenten attraktiv. Sprachbefehle wie „Alexa, spiele Musik“ oder „Siri, stelle einen Timer“ lassen sich intuitiv formulieren und bieten eine unkomplizierte Alternative zur Bedienung über komplexe Benutzeroberflächen. Sie können zudem freihändig und nebenbei, etwa beim Kochen oder Sport treiben, erteilt werden. Solche Sprachdialogsysteme und Gerätesteuerungen, aber auch Diktiersysteme, zählen zu den typischen Anwendungsbereichen der Spracherkennung (Euler, 2006, S. 17).

Prinzipiell lassen sich Spracherkennungssysteme unter verschiedenen Gesichtspunkten unterscheiden. Die Anpassungsfähigkeit an den Sprecher ist eine Möglichkeit. Unterschieden wird dabei zwischen sprecherunabhängigen Spracherkennungssystemen, bei welchen Äußerungen eines beliebigen Sprechers ohne zusätzliches Training erkannt werden, und sprecherabhängigen Spracherkennungssystemen, die speziell für einen bestimmten Sprecher trainiert werden. Je größer die Anzahl der berücksichtigten Sprecher, desto stärker wird die Variation in den Sprachaufnahmen und desto schwieriger wird die Erkennungsaufgabe. Zudem können Spracherkennungssysteme

nach dem Umfang der zu erkennenden Äußerung, zum Beispiel einzelne Wörter oder - weitaus schwieriger - fließend gesprochene Sätze, sowie der Größe des zu erkennenden Vokabulars unterschieden werden.

In dieser Arbeit wird die sprecherunabhängige Einzelworterkennung betrachtet, bei welcher davon ausgegangen wird, dass eine Äußerung nur ein einzelnes Wort umfasst. Dieses Wort ist Teil eines kleinen vorgegebenen Vokabulars und es können nur Wörter aus dem Vokabular erkannt werden. Die Erkennung erfolgt sprecherunabhängig mittels eines Mustervergleiches. Die Einzelworterkennung mittels Mustervergleich beinhaltet die Erstellung eines gesprochenen Musters für jedes Wort im Vokabular. Bei der Erkennung eines gesprochenen Wortes wird dieses mit allen verfügbaren Mustern verglichen. Als das erkannte Wort gilt jenes, dessen Muster am besten mit dem zu erkennenden Sprachsignal übereinstimmt.

Dieser Teil der Arbeit widmet sich einer stoffdidaktischen Analyse des Themas Einzelworterkennung, welche in der Konzipierung und Erprobung von Lehr- und Lernmaterial für einen Modellierungstag zum Thema Spracherkennung mündet. Kapitel 17 legt hierfür zunächst die mathematischen Grundlagen der Erstellung eines Einzelworterkenners mittels Mustervergleich dar und arbeitet die zentralen mathematischen Konzepte heraus. Die verwendeten Methoden werden didaktisch reduziert und als ein Modellierungsworkshop in Form von digitalem Lernmaterial umgesetzt, dessen Umsetzung in Kapitel 19 erläutert wird. Kapitel 20 zeigt auf, welche Erfahrungen bisher mit dem Einsatz der Lernmaterialien gemacht wurden und welche Anpassungen am Material daraufhin vorgenommen wurden. Es wird außerdem erläutert, wie die in Kapitel 7 vorgestellten Design-Prinzipien im Lernmaterial umgesetzt werden und damit insbesondere dargelegt, weshalb sich die Einzelworterkennung als Problemstellung für den mathematischen Modellierungsunterricht eignet.

17 Theoretischer Hintergrund

In diesem Kapitel werden die theoretischen Grundlagen der Einzelworterkennung mittels Mustervergleich genauer erläutert. Die Reduktion auf die Einzelworterkennung statt der Erkennung von fließend gesprochener Sprache ermöglicht die Konzentration auf die wesentlichen Verarbeitungsschritte der Spracherkennung, wie sie auch in komplexeren Spracherkennungssystemen Einsatz finden und erlaubt dennoch die Konstruktion eines einsatzfähigen Spracherkennungssystems mit den grundlegenden Methoden.

Die Verarbeitungsschritte und ihre Reihenfolge sind in Abbildung 17.1 dargestellt. Nachdem das Sprachsignal eingelesen beziehungsweise aufgenommen wurde, geht es im zweiten Schritt vorrangig um dessen Digitalisierung und Aufbereitung. Die digitalen Sprachsignale liegen im Zeitraum vor und können mit der Fourier-Transformation in ein Amplitudenspektrum umgewandelt werden. Anschließend werden typische Merkmale im Spektrum identifiziert. Hierfür werden einzelne Frequenzen zu Frequenzbändern unterschiedlicher Breite zusammengefasst. Die Merkmale werden für jeden Zeitblock als Merkmalsvektor abgespeichert. Alle Merkmalsvektoren eines Sprachsignals können anschließend zu einem Muster zusammengefasst werden. Diese Merkmalsextraktion wird für die Entwicklung eines Spracherkennungssystems zunächst für Sprachsignale durchgeführt, bei welchen die Wortbedeutung bekannt ist. Die ermittelten Muster werden als sogenannte *Referenzmuster* gemeinsam mit der Wortbedeutung abgespeichert und zu Klassen gleicher Wortbedeutung zusammengefasst. In der Anwendung des Spracherkennungssystems wird für ein noch unbekanntes Sprachsignal ebenfalls dessen Muster ermittelt. Anschließend wird das Sprachmuster mit jedem der Referenzmuster verglichen. Hierzu findet zunächst eine dynamische Zeitanpassung an das jeweilige Referenzmuster statt. Für die Klassifikation wird der Abstand vom angepassten Muster zum Referenzmuster bestimmt (Pfister & Kaufmann, 2017, S. 325). Das Sprachsignal wird der Klasse mit dem kleinsten Abstand zugeordnet.

Im Folgenden wird zunächst in Abschnitt 17.1 die Extraktion von Merkmalen aus einem Sprachsignal sowie deren Zusammenführung zu einem Muster thematisiert. Dabei wird die für diesen Prozess eine wichtige Rolle spielende Fouriertransformation aus Gründen der Vollständigkeit kurz rekapituliert. Eine vertiefte Behandlung erfolgt im Rahmen dieser Arbeit nicht, da an dieser Stelle auf bereits vorhandenes Lernmaterial zurückgegriffen wird (vgl. Abschnitt 19.2). Für eine eingehendere Auseinandersetzung mit der Theorie und Anwendung der Fouriertransformation wird auf die Werke Hoffmann und Wolff (2014) und Dahmen und Reusken (2008) verwiesen. Danach folgt in Abschnitt 17.2 die Klassifikation eines neuen Sprachmusters. Eine mögliche Verbesserung des entwickelten Modells wird in Abschnitt 17.3 vorgestellt.

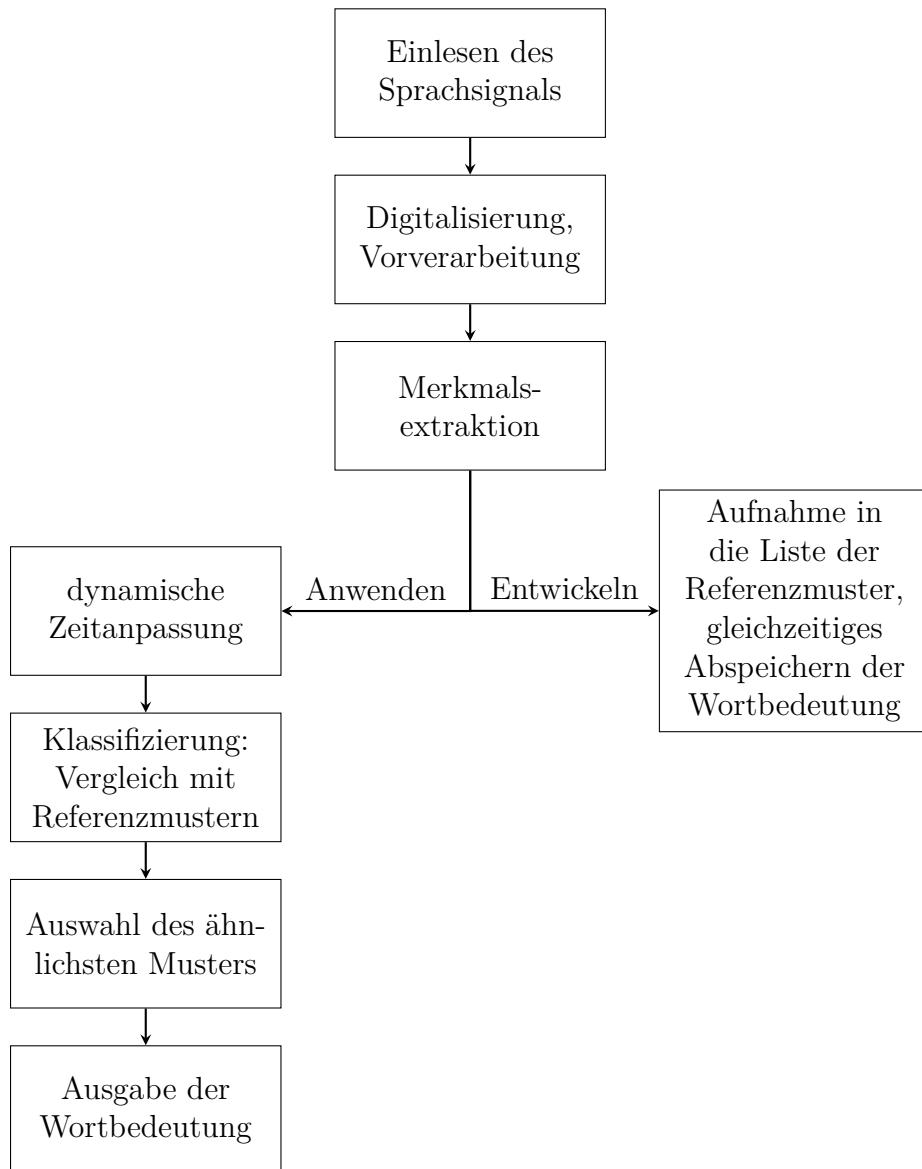


Abbildung 17.1: Verarbeitungsschritte bei der Einzelworterkennung, angelehnt an (Fellbaum, 2012, S. 312)

In Abschnitt 17.4 werden abschließend die Testphase beschrieben sowie verschiedene Qualitätsmaße eingeführt, mit denen eine Bewertung des Spracherkennungssystems vorgenommen werden kann. Es sei darauf hingewiesen, dass für die Erarbeitung der mathematischen Inhalte aufgrund eines Mangels an spezifischer mathematischer Fachliteratur teilweise auch auf Literatur aus angrenzenden Fachgebieten zurückgegriffen wird.

17.1 Vorverarbeitung und Merkmalsextraktion

17.1.1 Fouriertransformation

Die vom Menschen beim Sprechen erzeugten Töne sind Schallwellen. Diese können für die Spracherkennung über ein Mikrofon in ein zeitabhängiges elektrisches Signal umgewandelt werden. Dazu wird der Schalldruck zu bestimmten Zeitpunkten gemessen. Das digitalisierte Audiosignal x ergibt sich aus der Folge von Abtastwerten als zeitdiskrete Funktion des Schalldrucks mit

$$x : \mathcal{T} \rightarrow \mathcal{X}, \quad \mathcal{T} \subseteq \mathbb{Z}, \quad \mathcal{X} \subseteq \mathbb{R}.$$

Der Definitionsbereich \mathcal{T} steht hierbei für die Zeit, der Wertebereich \mathcal{X} für die Menge aller möglichen Signalwerte. Damit ein analoges Signal aus der Abtastfolge genau rekonstruiert werden kann, muss ein sich schnell änderndes Signal häufiger abgetastet werden. Diese Idee wird im *Abtasttheorem* genauer beschrieben: Es können nur solche Töne erfasst werden, deren Frequenz nur maximal halb so groß, wie die Abtastfrequenz ist, andernfalls können sie als ein Ton niedrigerer Frequenz fehlklassifiziert werden, und es kommt zum sogenannten *Aliasing* (Shannon, 1949, S. 12).

Besonders einfach darzustellende akustische Signale sind solche, die periodisch sind und nur eine geringe Anzahl an verschiedenen Frequenzen enthalten, wie beispielsweise ein Dreiklang. Ein Dreiklang entsteht durch das gleichzeitige Erklingen dreier verschiedener Töne mit festgelegten Intervallabständen. Diese Klänge lassen sich mathematisch als Summe dreier passender Sinusfunktionen

$$x_D(t_j) := \sin(2\pi f_a t_j) + \sin(2\pi f_b t_j) + \sin(2\pi f_c t_j)$$

ausdrücken. Hierbei wird das Signal an N äquidistanten Zeitpunkten t_1, \dots, t_N abgetastet.

Die Extraktion von Merkmalen erfolgt üblicherweise ähnlich zum menschlichen Hörorgan im Spektralbereich. Hierfür wird das Signal mit der Diskreten Fouriertransformation in den Frequenzbereich transformiert. Die diskrete Fouriertransformierte eines periodischen Signals, das an N äquidistanten Zeitpunkten t_1 bis t_N abgetastet wurde, lautet

$$X(f_i) := \frac{1}{N} \sum_{j=1}^N x(t_j) e^{-i2\pi f_i t_j}.$$

Dabei beschreiben f_i , mit $i = 1, \dots, N$, alle Frequenzen, nach denen im Signal gesucht wird (Hoffmann & Wolff, 2014, S. 116).

Betrachtet man lediglich ungerade periodische Funktionen, wie beispielsweise einen Dreiklang, so lässt sich die diskrete Fouriertransformation zu

$$X(f_i) = \frac{-2i}{N} \sum_{j=1}^C x(t_j) \sin(2\pi f_i t_j), \quad C = \begin{cases} \frac{N-1}{2}, & N \text{ ungerade} \\ \frac{N}{2} - 1, & N \text{ gerade} \end{cases}$$

vereinfachen (Hoffmann & Wolff, 2014, S. 168).

Kernaspekte: Zentrale mathematische Konzepte, die für die Modellierung von Tönen relevant sind, sind somit trigonometrische Funktionen. Für die Fouriertransformation ist die Vorstellung der Zerlegung einer Funktion in einzelne Sinusfunktionen unterschiedlicher Frequenzen zentral.

17.1.2 Kurzzeitspektralanalyse

Da es sich bei einem Sprachsignal nicht um ein zeitlich konstantes Signal handelt, sondern sich die Zusammensetzung der Frequenzen im Signal mit der Zeit ändert, macht es wenig Sinn, die Fouriertransformation für das gesamte Sprachsignal durchzuführen. Stattdessen wird das Sprachsignal in kleine Zeitblöcke unterteilt und die Fouriertransformation für jeden Zeitblock durchgeführt. Die Auswahl eines Zeitblocks des Signals erfolgt dabei über eine sogenannte *Fensterfunktion*. Wird dieses Fenster jedoch lediglich über eine Rechteckfunktion ausgewählt, so kann es zum unerwünschten *Leck-Effekt* kommen (Brigola, 2012): Bei einer Anwendung einer Rechteckfunktion kommt es zu einem Sprung im Signal zwischen dem Anfang des neuen Zeitblocks und dem Ende des letzten Zeitblocks. Dieser Sprung wird bei der Anwendung der Fouriertransformation mit analysiert, es kommt zu Überschwingungen nahe der Unstetigkeitsstelle, dem sogenannten *Gibbschen Phänomen* (Hewitt & Hewitt, 1979), und es werden schließlich Frequenzen detektiert, die im ursprünglichen Signal nicht enthalten waren. Um diesen Effekt zu vermindern, wird das in der Sprachverarbeitung am häufigsten genutzte Fenster (Pfister & Kaufmann, 2017, S. 66) – das Hamming-Fenster

$$w(t_j) := c_0 - (1 - c_0) \cos\left(\frac{2\pi t_j}{t_N}\right), \quad 1 \leq j \leq N, \quad c_0 \in [0.5, 1] \quad (17.1)$$

verwendet (Hamming, 1989, S. 119), dessen Ränder weniger steil abfallen als bei einer Rechteckfunktion. Dabei handelt es sich außerdem im Vergleich zur Rechteckfunktion um eine glatte Funktion.

Mathematisch stellt das Auswählen eines Zeitblockes eines Signals die Multiplikation des Signals mit einer Fensterfunktion dar. Die gefensterte Fouriertransformation für das p -te Fenster ergibt sich somit zu

$$X_p(f_i) = \frac{1}{N} \sum_{j=1}^N x_p(t_j) w(t_j) e^{-i2\pi f_i t_j},$$

wobei x_p das Signal im p -ten Fenster bezeichnet. Die Fouriertransformierte des kompletten Signals, das sogenannte Frequenzspektrum, kann somit als Matrix

$$\begin{pmatrix} X_1(f_1) & \cdots & X_P(f_1) \\ \vdots & & \vdots \\ X_1(f_N) & \cdots & X_P(f_N) \end{pmatrix}$$

dargestellt werden, wobei P die Anzahl der Fenster und N die Anzahl der untersuchten Frequenzen darstellt.

Kernaspekte: Zentral ist somit die Unterteilung eines nicht periodischen Signals in Abschnitte für die Fourier-Analyse und damit mathematisch gesehen die Multiplikation des Signals mit einer geeigneten Fensterfunktion. Vereinfachend kann statt der genauen Form der Fensterfunktion lediglich die Unterteilung in Blöcke thematisiert werden.

17.1.3 Filterbank

Für die Weiterverarbeitung wird die große Datenmenge aus dem Frequenzspektrum mit verschiedenen Methoden reduziert, um den Verarbeitungsaufwand zu minimieren und die Redundanz, welche die Analyse und Klassifikation des Sprachsignals behindert, zu eliminieren. Das Ziel besteht darin, nur die relevanten Merkmale aus den Daten zu extrahieren (Fellbaum, 2012, S. 251).

Für die Reduktion der Daten kann sich am menschlichen Gehör, das vermutlich auf die Wahrnehmung von Sprache optimiert ist, orientiert werden. Aus der Psychoakustik ist bekannt, dass Frequenzen, die sehr nahe beieinander liegen, vom menschlichen Gehör als ein Ton aufgefasst werden (Euler, 2006, S. 35). Die Frequenzen können daher in Gruppen zu sogenannten Frequenzbändern zusammengefasst werden.

Zudem wurde festgestellt, dass die Frequenzauflösung des menschlichen Gehörs mit höheren Frequenzen stark abnimmt. Die Breite der Frequenzbänder nimmt somit bei hohen Frequenzen zu (Euler, 2006, S. 36). Zwischen der subjektiv wahrgenommenen Tonhöhe und der Frequenz ergibt sich der logarithmische Zusammenhang (Fellbaum, 2012, S. 94)

$$m(f) := 2595 \cdot \log_{10} \left(1 + \frac{f}{700 \text{ kHz}} \right), \quad (17.2)$$

der als *Mel-Skala* bezeichnet und in der Einheit mel gemessen wird (Euler, 2006, S. 35). Als Referenzpunkt dient der Ton mit einer Frequenz von 1000 Hz, welchem auf der Mel-Skala ein Wert von 100 mel zugeordnet wird (Pfister & Kaufmann, 2017, S. 95).

Die ermittelten Frequenzen werden über die Mel-Skala skaliert, und anschließend werden alle Frequenzen eines Frequenzbandes mit einer vorgegebenen Intervalllänge δ_m zu einer sogenannten *Mittenfrequenz* zusammengefasst. Hierzu werden typischerweise überlappende dreieckige Filter verwendet (vgl. Abbildung 17.2). Jeder Dreiecksfilter nimmt bei seiner Mittenfrequenz den maximalen Wert an und fällt bei der Mittenfrequenz zweier benachbarter Filter linear auf null ab. Die Überlappung der Filter sorgt dafür, dass dem Einfluss einer zufällig ausgewählten Intervallgrenze entgegen gewirkt werden kann.

Für Sprachsignale wird zusätzlich empfohlen, die Frequenzen außerhalb eines Intervall, begrenzt von einer unteren Grenzfrequenz f_{min} von 100 Hz und einer oberen

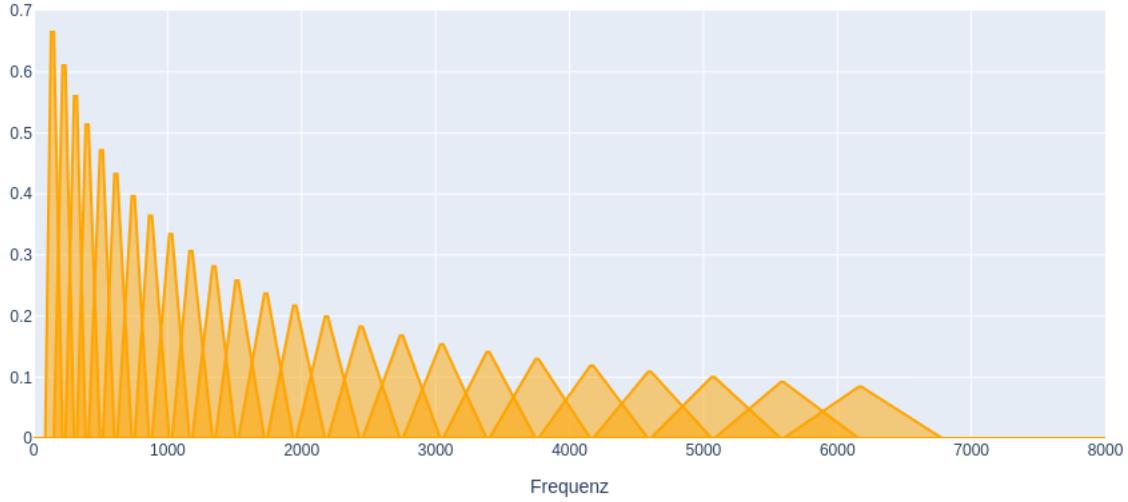


Abbildung 17.2: Mel-Filterbank mit 25 Dreiecksfiltern, einer unteren Grenzfrequenz von 100 Hz und einer oberen Grenzfrequenz von 6800 Hz

Grenzfrequenz f_{max} von 6800 Hz für die Merkmalsbildung außen vorzulassen (M & Kopparapu, 2014, S. 4). Der Grund hierfür liegt im menschlichen Hörmodell, das zeigt, dass Sprachsignale nur zwischen diesen beiden Grenzfrequenzen wahrgenommen werden können. Signale, die außerhalb dieser Grenzfrequenzen liegen, sind Hintergrundgeräusche, die aus dem Signal entfernt werden können.

Die Frequenzauflösung kann bei einer vorgegebenen Anzahl von Q Filtern über die Formel

$$\delta_m := \frac{m(f_{max}) - m(f_{min})}{Q + 1}$$

bestimmt werden. Die Lage der q -ten Mittenfrequenzen ist dabei über

$$m(f_{mitten}(q)) := q \cdot \delta_m$$

wobei $q = 1, 2, \dots, Q$ gegeben. Um die Lage der Mittenfrequenzen in Hertz zu erhalten, kann die Inverse von Gleichung (17.2) angewendet werden, und es ergibt sich

$$f_{mitten}(q) := 700 \cdot (10^{\frac{m(f_{mitten}(q))}{2595}} - 1).$$

Die überlappenden Dreiecksfilter sind schließlich definiert als

$$D(f_i, q) := \begin{cases} 0 & , f_i < f_{mitten}(q-1) \\ 100 \cdot \frac{f_i - f_{mitten}(q-1)}{(f_{mitten}(q) - f_{mitten}(q-1))^2} & , f_{mitten}(q-1) \leq f_i < f_{mitten}(q) \\ 100 \cdot \frac{f_i - f_{mitten}(q+1)}{(f_{mitten}(q) - f_{mitten}(q+1))^2} & , f_{mitten}(q) \leq f_i < f_{mitten}(q+1) \\ 0 & , f_i \geq f_{mitten}(q+1) \end{cases}.$$

Die Gesamtheit dieser Dreiecksfilter stellt die *Filterbank* dar. Die Anwendung der Filterbank entspricht einer Multiplikation der Dreiecksfilter mit der Fouriertransformierten eines Fensters. Nach Anwenden der Filterbank ergibt sich

$$X'_p(q) := \sum_{i=1}^N X_p(f_i) \cdot D(f_i, q).$$

Die Funktion $X'_p(k)$ beschreibt die Ausprägung zur q -ten Mittenfrequenzen (im p -ten Fenster). Diese Ausprägungen werden anschließend als Merkmale für die Klassifikation verwendet. So kann mit einer überschaubaren Anzahl an Merkmalen weitergearbeitet werden. Die ermittelten metrisch skalierten Merkmale $X'_p(1), \dots, X'_p(Q)$ werden schließlich pro Fenster zu einem Merkmalsvektor \vec{a} zusammengefasst.

Kernaspekte: Um das Signal zu filtern, werden psychoakustische Phänomene mathematisch modelliert. Letztlich entspricht die Filterung der Bildung eines gewichteten Mittelwertes.

17.2 Klassifikation

17.2.1 Abstandsmaß

Um die Ähnlichkeit der Sprachmuster festzustellen, kann die Nähe der Merkmalsvektoren betrachtet werden. Hierzu wird zunächst ein Abstandsmaß festgelegt. Sind für das Abstandsmaß, welches eine Abbildung $d_M : M \times M \rightarrow \mathbb{R}$ ist, die Aussagen

$$d_M(u, v) \geq 0 \text{ und } d_M(u, v) = 0 \Leftrightarrow u = v \text{ (positive Definitheit)}, \quad (17.3)$$

$$d_M(u, v) = d_M(v, u) \text{ (Symmetrie)}, \quad (17.4)$$

$$d_M(u, v) \leq d_M(u, w) + d_M(w, v) \text{ (Dreiecksungleichung)} \quad (17.5)$$

für u, v und w aus M erfüllt, so wird dieses als *Metrik* auf M bezeichnet (Frochte, 2021, S. 110). Eine Metrik auf M kann über

$$d_M(u, v) = \|u - v\| \quad (17.6)$$

auch durch eine Norm $\|\cdot\|$ auf M induziert werden (Frochte, 2021, S. 110).

Ein typisches Beispiel für eine Metrik ist die euklidische Metrik, welche auch für die Klassifizierung der Sprachmuster eingesetzt wird. Die *euklidische Metrik* wird von der auch als euklidische Norm bezeichneten ℓ^2 -Norm

$$\|\vec{v}\|_2 := \sqrt{\sum_{i=1}^n v_i^2}, \quad (17.7)$$

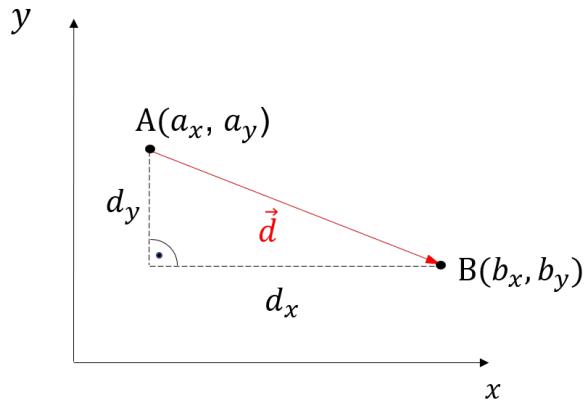


Abbildung 17.3: Euklidischer Abstand zwischen den Punkten A und B im zweidimensionalen Raum

induziert, und sie ist definiert durch

$$d(\vec{v}, \vec{w}) := \sqrt{\sum_{i=1}^n (v_i - w_i)^2}. \quad (17.8)$$

Dabei sind \vec{v} und \vec{w} zwei Vektoren aus \mathbb{R}^n mit Einträgen v_i , $i = 1, \dots, n$, und w_i , $i = 1, \dots, n$. Die euklidische Metrik ist anschaulich die Länge beziehungsweise der Betrag des Verbindungsvektors \vec{d} der Vektoren \vec{v} und \vec{w} . Abbildung 17.3 zeigt den mit der euklidischen Metrik berechneten und auch *euklidischer Abstand* genannten Abstand der (Ortsvektoren der) beiden Punkte $A(a_x, a_y)$ und $B(b_x, b_y)$, im zweidimensionalen Fall. Bei der Berechnung des euklidischen Abstands für den zweidimensionalen Fall ergibt sich der Satz des Pythagoras als bekannter Spezialfall.

17.2.2 Abstände von Sprachmustern

Es werden nun die beiden Sprachsignale x_a und x_b und deren Sprachmuster betrachtet. Die Sprachmuster ergeben sich als Aneinanderreihung der Merkmalsvektoren, deren Ermittlung in Kapitel 17.1 beschrieben ist. Im Folgenden werden diese Sprachmuster als

$$A = (\vec{a}_1, \vec{a}_2, \dots, \vec{a}_P) \quad \text{mit } \vec{a}_p = \begin{pmatrix} X'_{a,p}(1) \\ \vdots \\ X'_{a,p}(K) \end{pmatrix}$$

und

$$B = (\vec{b}_1, \vec{b}_2, \dots, \vec{b}_P) \quad \text{mit } \vec{b}_q = \begin{pmatrix} X'_{b,q}(1) \\ \vdots \\ X'_{b,q}(K) \end{pmatrix}$$

bezeichnet. Um die Unterschiede im Sprachverlauf festzustellen, wird der euklidische Abstand

$$d(\vec{a}_p, \vec{b}_q) = \sqrt{\sum_{i=1}^K [X'_{a,p}(i) - X'_{b,q}(i)]^2} \quad (17.9)$$

der Merkmalsvektoren \vec{a}_p und \vec{b}_q in jedem Zeitblock bestimmt. Je kleiner dieser lokale Abstand ist, desto stärker spricht das dafür, dass an der Stelle p im Muster A und der Stelle q im Muster B derselbe Laut vorliegt.

Andere Abstandsmaße wie die Manhattan-Metrik als absolute Abweichung

$$d_{abs}(\vec{a}_p, \vec{b}_q) := \sum_{i=1}^K |X'_{a,p}(i) - X'_{b,q}(i)| \quad (17.10)$$

können ebenfalls als Maß für die Ähnlichkeit herangezogen werden. Die Manhattan-Metrik, welche von der ℓ^1 -Norm induziert ist, gewichtet Ausreißer weniger stark als dies bei der euklidischen Metrik der Fall ist. Denkbar ist es auch die Kosinus-Ähnlichkeit als Maß für die Ähnlichkeit heranzuziehen. Sie misst die Übereinstimmung der Richtung im euklidischen Raum und lässt im Vergleich zur euklidischen Metrik und zur Manhattan-Metrik die Länge der Vektoren unberücksichtigt (Sartorius, 2018, S. 122 - 124). Sie ist allerdings keine Metrik, da sie die Dreiecksungleichung nicht erfüllt, wird aber aufgrund der genannten Eigenschaft trotzdem im Zusammenhang mit dem Nächster-Nachbar-Verfahren verwendet (Cunningham & Delany, 2021, S. 3 - 4; Sartorius, 2018, S. 113).

Die Wahl eines sinnvollen Abstandsmaßes ist eine interessante Modellentscheidung, die insbesondere mit Blick auf den realen Kontext und der Verwendung als Modellparameter beim Nächster-Nachbar-Verfahren getroffen werden muss (vgl. Abschnitt 17.2.3). Im Hinblick auf eine möglichst einfache Darstellung, die das Vorwissen der Schülerinnen und Schüler zum Abstandsbegriff aufgreift, wird hier die Entscheidung für die euklidische Metrik getroffen (vgl. Abschnitt 19.2.4).

Vergleicht man die Merkmalsvektoren nun Zeitfenster für Zeitfenster mithilfe des euklidischen Abstands, so kann der Gesamtabstand $D(A, B)$ als Summe der lokalen Abstände $d(\vec{a}_p, \vec{b}_q)$ zu

$$D(A, B) := \sum_{p,q=1}^P d(\vec{a}_p, \vec{b}_q) \quad (17.11)$$

berechnet werden. Der Gesamtabstand erfüllt als Summe einzelner Metriken alle Eigenschaften einer Metrik und ist somit selbst ebenfalls eine Metrik. Wird daselbe Wort zweimal hintereinander mit identischer Geschwindigkeit gesprochen und die Aufnahme beginnt und endet jeweils zum gleichen Zeitpunkt, so kann der Gesamtabstand als Maß für die Verschiedenheit der beiden Sprachsignale dienen. Wird der Abstand über die euklidische Metrik bestimmt, so wird somit die Annahme getroffen, dass die genannten Bedingungen erfüllt sind. Im weiteren Verlauf wird gezeigt, dass es auch eine Möglichkeit gibt, diese Unterschiede im zeitlichen Verlauf

der Sprachsignale zu berücksichtigen. Die entsprechende Methode wird in Abschnitt 17.3 ausführlich behandelt.

Sollen Sprachsignale klassifiziert werden, deren Wortbedeutung noch unbekannt ist, so muss zunächst für Sprachsignale mit bekannter Wortbedeutung das Sprachmuster ermittelt werden. Diese Muster werden auch als *Referenzmuster* R_1, \dots, R_r bezeichnet und können in Klassen $1, \dots, K$ mit gleicher Wortbedeutung abgespeichert werden. Dieser Schritt stellt das *Training* des Modells dar (vgl. Abschnitt 3.2). Als Trainingsdaten wird das Tupel $(R_i, W^{R_i}), i = 1, \dots, r$, mit den Referenzmustern R_i und der zugehörigen Wortbedeutung W^{R_i} mit $W^{R_i} \in \{1, \dots, K\}$ bezeichnet. Um ein Sprachsignal, dessen Wortbedeutung unbekannt ist, zu erkennen, wird zunächst dessen Sprachmuster T_j ermittelt. Dieses soll nun einer Klasse zugeordnet und damit dessen vorhergesagte Klassenzugehörigkeit und Wortbedeutung W^{T_j} ermittelt werden. Ausgangspunkt ist somit ein Mehrklassen-Klassifikationsproblem, wobei die Klassen die Referenzmuster mit gleicher Wortbedeutung darstellen.

17.2.3 Nächster-Nachbar-Verfahren

Als Klassifizierungsverfahren kann das Nächster-Nachbar-Verfahren (engl. Nearest-Neighbour-Method) (Fellbaum, 2012, S. 273) verwendet werden. Dabei handelt es sich um ein sehr verbreitetes nichtparametrisches Verfahren (Botsch, 2023, S. 103), das unter anderem zur Lösung von Klassifizierungsproblemen eingesetzt wird. Nichtparametrisches Verfahren bedeutet, dass keine Verteilungsannahme über die Daten in der Grundgesamtheit getroffen wird. Beim Nächster-Nachbar-Verfahren werden neue Datenpunkte unter Berücksichtigung der Klassenzugehörigkeit des nächstliegenden Datenpunktes klassifiziert. Um die Nähe zweier Datenpunkte zu bestimmen, wird eine Metrik d_M auf der Menge M gewählt.¹ Der nächstliegende Datenpunkt einer Menge aus Punkten $\{y_1, \dots, y_I\} \subseteq M$ zu einem gegebenen Punkt $x \in M$, auch *nächster Nachbar* genannt, ist durch

$$\underset{y_i, i=1, \dots, I}{\operatorname{argmin}} d_M(x, y_i) \quad (17.12)$$

gegeben. Soll ein neuer Datenpunkt klassifiziert werden, so wird dessen nächster Nachbar bestimmt. Der neue Datenpunkt wird schließlich der Klasse des nächsten Nachbarn zugeordnet.

Dieses Verfahren ist ein Spezialfall des k -nächste-Nachbarn-Verfahrens, bei welcher die Klassenzugehörigkeit der k nächsten Nachbarn ausschlaggebend für die Klassifizierung eines neuen Datenpunktes ist. Die Auswahl der Klasse des neuen Datenpunktes erfolgt hier durch eine Mehrheitsentscheidung. Je größer der Parameter k gewählt wird, desto stärker generalisiert das Klassifizierungsverfahren, je kleiner er gewählt wird, desto stärker fallen Ausreißer ins Gewicht. Da für das vorliegende

¹Es ist generell möglich, für das Nächster-Nachbar-Verfahren ein Ähnlichkeitsmaß zu wählen, welches keine Metrik ist. Es gibt jedoch einige Leistungsoptimierungen des grundlegenden Verfahrens, welche die Verwendung einer Metrik erfordern, weshalb in der Praxis zumeist Metriken verwendet werden (Cunningham & Delany, 2021).

Beispiel der Klassifizierung der Sprachsignale die Betrachtung des nächsten Nachbarn als ausreichend erachtet wird, wird auf eine ausführlichere Betrachtung des k -nächste-Nachbarn-Verfahren verzichtet.

Das k -nächste-Nachbarn-Verfahren zählt aufgrund der Tatsache, dass die Hauptarbeit nicht beim Training sondern bei der Abfrage stattfindet, als sogenanntes *Lazy Learning* (Frochte, 2021). Konkret bedeutet dies, dass der wesentliche Rechenaufwand in der Berechnung der Abstände zwischen einem neuen Datenpunkt und seinen Nachbarn liegt. Da diese Berechnung erst im Moment der Klassifizierung erfolgt, muss sie für jeden neu zu klassifizierenden Datenpunkt erneut durchgeführt werden.

Wird das Nächster-Nachbar-Verfahren für die Spracherkennung angewendet, so bedeutet dies, dass ein neues Sprachsignal aufgrund seiner Ähnlichkeit zu den Referenzmustern einer Klasse zugeordnet wird. Die Datenpunkte sind hier die Sprachmuster. Soll ein Testmuster T_j klassifiziert werden, so wird der Abstand $D(T_j, R_i)$ zwischen jedem Referenzmuster R_i und dem zeitlich an das Referenzmuster angepassten Testmuster T_j bestimmt. Als Abstandsmetrik wird die Summe der euklidischen Abstände verwendet (vgl. (17.9)). Ausgewählt wird die Klasse mit Wortbedeutung W^R , die das Referenzmuster enthält, welches den kleinsten Abstand zum Sprachmuster aufweist. Die Wortbedeutung dieser Klasse wird als das erkannte Wort $W_R^{T_j}$ mit

$$W_R^{T_j} := W^R, \quad R = \operatorname{argmin}_{R_i, i=1, \dots, r} D(T_j, R_i) \quad (17.13)$$

festgelegt. Die Funktion in (17.13) dient somit als Zuordnungsfunktion der Muster T_j zu einer Klasse, in diesem Fall der Wortbedeutung (vgl. überwachtes Lernen in Abschnitt 3.2).

Kernaspekte: Zentrale mathematische Konzepte in diesem Kapitel sind Abstandsmaße zwischen Punkten im Mehrdimensionalen sowie das Konzept der Klassifizierung und die Rolle der Trainingsdaten, den diese dabei innehaben. Beim Nächster-Nachbar-Verfahren als hier ausgewähltes Klassifizierungsverfahren ist das Formulieren eines Optimierungsproblems zentral. Dieses wird durch den Vergleich aller Abstände gelöst.

Neben dem hier aufgeführten Nächster-Nachbar-Verfahren werden in der Literatur weitere Verfahren zur Klassifikation von Sprachsignalen angeführt. Zu den wichtigsten zählen (Fellbaum, 2012, S. 259 - 260)

- statistische Klassifikationsverfahren wie der Naive Bayes-Klassifikator,
- Hidden Markov-Modelle,
- Lineare Klassifikationsverfahren (Klassifikationsverfahren, bei denen lineare Trennfunktionen verwendet werden),
- Neuronale Netze.

Fellbaum (2012) nennt außerdem die sogenannte *dynamische Zeitanpassung* als wichtiges Verfahren für die Klassifizierung von Sprachsignalen. Hierbei handelt es sich allerdings nicht direkt um ein Klassifikationsverfahren, sondern um eine vorbereitende Maßnahme vor der eigentlichen Klassifizierung, welche das zu erkennende

Sprachsignal durch eine dynamische Zeitanpassung an das Referenzsignal anpasst. Dieses Verfahren wird im nächsten Abschnitt als Verbesserung des bisherigen Modells vorgestellt.

17.3 Modellverbesserung: zeitliche Anpassung der Sprachmuster

Werden in der Praxis unterschiedliche Sprachaufnahmen desselben Wortes verglichen, so lassen sich beträchtliche Unterschiede im Signalverlauf feststellen. Zum einen unterscheidet sich die Dauer der Signale, und auch die Länge der einzelnen Laute variiert stark. So kann das Wort „Banane“ auch mit einem langen ersten „a“ also als „Baaanane“ ausgesprochen werden. Trotzdem ist damit dasselbe Wort gemeint. Diese Unterschiede im Sprachverlauf sollen im Folgenden berücksichtigt werden. Dazu wird die Folge der Merkmalswerte als Zeitreihe betrachtet.

Um die genannten Unterschiede im Signalverlauf der Sprachmuster bei der Erkennung auszugleichen, werden die Sprachmuster zeitlich aufeinander angepasst, sodass zueinander passende Bereiche der Muster übereinander liegen. Da die zeitlichen Verschiebungen und Verzerrungen nicht nur die gesamte Äußerung, sondern wie oben im Beispiel auch nur Teile davon betreffen können, lässt sich eine Anpassung nicht einfach durch eine globale Stauchung oder Streckung des Sprachmusters erreichen. Vielmehr müssen die einzelnen Teile des Sprachmusters individuell durch kleine lokale Verzerrungen aufeinander angepasst werden. Ziel ist es, die Sprachmuster so durch lokale Verzerrungen aufeinander anzupassen, dass die Summe der lokalen Abstände möglichst klein wird. Dieser Vorgang wird als *dynamische Zeitanpassung* (engl. *Dynamic Time Warping* (DTW)) bezeichnet (Pfister & Kaufmann, 2017, S. 356). Das Verfahren ist nicht nur in der Spracherkennung bekannt, sondern wird auch im Allgemeinen beim Vergleich von Zeitreihen angewendet (Marteau, 2009, S. 306).

17.3.1 Zuordnungsfunktion

Die zeitliche Anpassung zweier Sprachmuster A und B kann über die Funktion

$$\phi(k) = (\phi_A(k), \phi_B(k))$$

mit $k = 1, \dots, T, T \in \mathbb{N}$, und $\phi_A(k), \phi_B(k) \in 1, \dots, P$ beschrieben werden. Sie ordnet die Zustände in Muster A den Zuständen in Muster B zu. Der Graph dieser *Zuordnungsfunktion* wird auch als *Zuordnungspfad* bezeichnet.

An die Zuordnungsfunktion werden verschiedene Randbedingungen gestellt (Euler, 2006, S. 57; Fellbaum, 2012, S. 296 - 297; Pfister & Kaufmann, 2017, S. 358).

- Kontinuität: Keine größeren Teile des Musters und damit z.B. ein ganzer Laut sollen übersprungen werden können. Dies resultiert in der Bedingung, dass

jedes Merkmal des Musters A und jedes Merkmal des Musters B angenommen werden soll. Es gilt

$$\phi_A(k+1) - \phi_A(k) \in \{0, 1\} \quad \text{und} \quad \phi_B(k+1) - \phi_B(k) \in \{0, 1\}. \quad (17.14)$$

- Randwerte: Beide Sprachsignale starten und enden zum gleichen Zeitpunkt. Die Zuordnung startet mit $(1, 1)$ und endet mit (P, P) . Somit gilt

$$\phi(1) = (1, 1) \quad \text{und} \quad \phi(T) = (P, P). \quad (17.15)$$

- Monotonie: Die einzelnen Laute des Sprachsignals folgen einem festen zeitlichen Ablauf, der beibehalten werden soll. Die Merkmale müssen daher in ihrer ursprünglichen Reihenfolge eingesetzt werden. Die Zuordnungsfunktion ϕ ist somit monoton wachsend. Es gilt

$$\phi_A(k+1) \geq \phi_A(k) \quad \text{und} \quad \phi_B(k+1) \geq \phi_B(k). \quad (17.16)$$

Modifikationen der Randbedingungen sind möglich und nicht unüblich. Die Kontinuitätsbedingung wird beispielsweise häufig aufgeweicht, und das Überspringen eines Merkmals wird zugelassen (Fellbaum, 2012, S. 297; Pfister & Kaufmann, 2017, S. 358). Für diese Arbeit soll die klassische Wahl der oben vorgestellten Randbedingungen genügen.

Bei gegebener Zuordnung kann wieder eine Art Gesamtabstand zwischen zwei Sprachmustern zu

$$D_\phi(A, B) = \sum_{k=1}^T d(A(\phi_A(k)), B(\phi_B(k))) \quad (17.17)$$

bestimmt werden. Zur eindeutigen Kennzeichnung der zugrundeliegenden Zuordnung wird der Gesamtabstand entsprechend indiziert.

Gesucht ist nun diejenige Zuordnung, die diesen Gesamtabstand der Sprachmuster minimiert, sodass sich das Optimierungsproblem

$$D_{DTW}(A, B) := \min_{\phi} \sum_{k=1}^T d(A(\phi_A(k)), B(\phi_B(k))) \quad (17.18)$$

als Minimierung der Summe der lokalen Abstände, hier beispielhaft mit der ℓ^2 -Norm formulieren lässt. Die Lösung des Optimierungsproblems existiert bei zwei gleichlangen Mustern, wovon hier stets ausgegangen wird, immer. Insbesondere ist die Identität in diesem Fall stets eine zulässige Zuordnungsfunktion, die die Randbedingungen erfüllt. Da die Menge der möglichen Zuordnungsfunktionen endlich ist, wird das Minimum stets angenommen. Häufig existieren allerdings mehrere Zuordnungspfade, die den Gesamtabstand minimieren. Die Lösung des Optimierungsproblems ist dann nicht eindeutig.

In der Literatur wird gelegentlich der Begriff *Dynamic Time Warping Distanz* verwendet (z.B. Jain, 2019). Allerdings stellt diese Form der Abstandsbestimmung im mathematischen Sinne keine Metrik dar. Bei der Wahl symmetrischer Randbedingungen, also solche, die an beide Muster die gleichen Bedingungen stellen, wie bei den oben beschriebenen Randbedingungen, ist die Symmetrieverbedingung (Bedingung (17.4)) erfüllt. Die positive Definitheit (Bedingung (17.3)) ist allerdings nicht erfüllt (Müller, 2021, S. 140 - 141; Vidal Ruiz et al., 1985, S. 334 - 335). Aus $D_{DTW}(A, B) = 0$ folgt beispielsweise nicht unbedingt, dass das Muster A dem Muster B entspricht. Die Dreiecksungleichung (Bedingung (17.5)) für D_{DTW} ist ebenfalls nicht erfüllt (Vidal Ruiz et al., 1985, S. 334 - 335). Die Konstruktion eines Gegenbeispiels lässt das schnell einsehen. Abbildung 17.4 zeigt ein solches Gegenbeispiel für drei konstruierte Zeitreihen A , B und C mit den vorgestellten symmetrischen Randbedingungen. Die Abbildung zeigt drei Graphen mit drei Zuordnungspfaden, bei denen jeweils unterhalb der x -Achse und links der y -Achse die Muster der anzupassenden Zeitreihen abgebildet sind.

Wird das Anpassen der Sprachmuster allerdings gedanklich vor die Bestimmung des Abstands gesetzt und $\tilde{\phi}$ als die optimale Zuordnung ermittelt, so kann der Abstand $D(A_{\tilde{\phi}}, B_{\tilde{\phi}})$ zwischen den angepassten Sprachmustern $A_{\tilde{\phi}}$ und $B_{\tilde{\phi}}$ als Summe der euklidischen Abstände der einzelnen Zeitblöcke bestimmt werden. Für diesen Abstand gilt

$$\begin{aligned} D(A_{\tilde{\phi}}, B_{\tilde{\phi}}) &= \sum_{k=1}^T d(A_{\tilde{\phi}}(k), B_{\tilde{\phi}}(k)) = \sum_{k=1}^T d(A(\tilde{\phi}_A(k)), B(\tilde{\phi}_B(k))) D_{\tilde{\phi}}(A, B) \\ &= D_{DTW}(A, B). \end{aligned}$$

Mit der Lösung des Optimierungsproblems wird somit gleichzeitig der Abstand der optimal aufeinander angepassten Sprachmuster bestimmt.

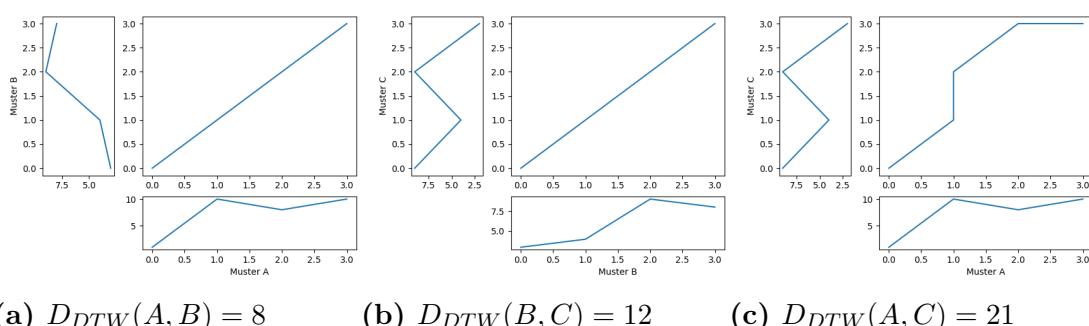


Abbildung 17.4: Verletzung der Dreiecksungleichung durch D_{DTW} am Beispiel der Sprachmustern $A = (1, 10, 8, 10)$, $B = (3, 4, 9, 8)$, $C = (9, 4, 9, 2)$

17.3.2 Lösen des Optimierungsproblems: dynamische Programmierung

Die naive Herangehensweise zur Lösung des Optimierungsproblems besteht darin, den Gesamtabstand für jede mögliche Zuordnung zu berechnen, was jedoch mit einem äußerst hohen Rechenaufwand verbunden ist. Stattdessen wird die Methode der dynamischen Programmierung genutzt, die ein komplexes Problem in voneinander unabhängige, leicht lösbare Teilprobleme zerlegt. Somit wird das ursprüngliche Problem gelöst, indem die Teilprobleme rekursiv gelöst und deren Lösungen zur späteren Verwendung gespeichert werden (Cormen, 2001, S. 362).

Im Falle des Optimierungsproblems in (17.18) wird derjenige Zuordnungspfad gesucht, welcher für den kleinsten Gesamtabstand zwischen den gemäß der Zuordnung angepassten Sprachmustern sorgt. Statt mit (17.17) den Gesamtabstand zwischen den Sprachmustern für jede mögliche Zuordnung zu berechnen, wird der Gesamtabstand in Teilsummen unterteilt, und es wird die Redundanz gemeinsamer Pfade ausgenutzt. Dafür wird jeder Zeitblock einzeln betrachtet, indem zuerst mögliche zeitliche Anpassungen der Sprachmuster in diesem Zeitblock in den Blick genommen werden und anschließend für jede Anpassung der Abstand der angepassten Sprachmuster bestimmt wird. Ausgewählt wird anschließend diejenige Zuordnung, die bis zu diesem Zeitblock den kleinsten Abstand zwischen den angepassten Sprachmustern aufweist.

Das Verfahren wird im Folgenden ausführlich beschrieben. Die Zwischenwerte des Gesamtabstands bis zu einem bestimmten Zeitblock p in A bzw. Zeitblock q in B werden dabei mit $D_{p,q}$ bezeichnet. Im ersten Schritt ergibt sich der Abstandswert aufgrund der Bedingung zu den Randwerten zu (vgl. Abschnitt 17.3.1)

$$D_{1,1} = d(A(1), B(1)).$$

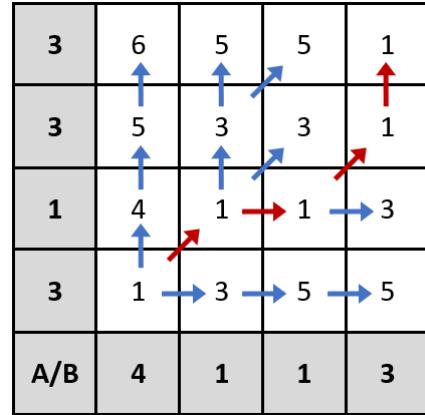
Die Wahl der Zuordnungsfunktion spielt hier noch keine Rolle. Für eine bessere Übersicht können die Abstandswerte $D_{p,q}$ für jede Zustandskombination der beiden Muster in den Feldern einer $P \times P$ großen Tabelle zwischengespeichert werden (siehe Abbildung 17.5). Die Werte der Muster A und B werden dabei unten und links an den Rändern der Tabelle angeordnet.

Im zweiten Schritt sind drei mögliche Erweiterungen des Zuordnungspfades möglich, woraus sich bereits die drei möglichen Zuordnungen $\phi(2) \in \{(1, 2), (2, 1), (2, 2)\}$ ergeben. In den ersten beiden Fällen ergibt sich der Abstandswert des zweiten Schrittes zu

$$\begin{aligned} D_{1,2} &= D_{1,1} + d(A(1), B(2)), \\ D_{2,1} &= D_{1,1} + d(A(2), B(1)). \end{aligned}$$

Komplizierter ist die Berechnung des Gesamtabstands $D_{2,2}$. Der Zustand $(2, 2)$ ist nämlich neben dem direkten diagonalen Weg auch über die Folge $(1, 1), (1, 2), (2, 2)$ und über die Folge $(1, 1), (2, 1), (2, 2)$ zu erreichen. In der Tabelle entsprechen die drei Möglichkeiten dem diagonalen Weg, dem Weg nach rechts und dann nach oben, sowie

A(P)	D _{P,1}	D _{P,2}	...	D _{P,P}
⋮
A(2)	D _{2,1}	D _{2,2}	...	D _{2,P}
A(1)	D _{1,1}	D _{1,2}	...	D _{1,P}
A/B	B(1)	B(2)	...	B(P)



3	6	5	5	1
3	5	3	3	1
1	4	1	1	3
3	1	3	5	5
A/B	4	1	1	3

(a) in der allgemeinen Form

(b) mit fiktiven Beispilmustern bestehend aus einem Merkmal pro Zeitblock

Abbildung 17.5: Tabelle für die Zwischenspeicherung der Abstandswerte

dem Weg nach oben und dann nach rechts. Weitere Möglichkeiten gibt es aufgrund der Einschränkungen durch die Randbedingungen nicht. Für den Gesamtabstand erhält man je nach Wahl der Zuordnung die drei Möglichkeiten

$$\begin{aligned}
 & D_{1,1} + d(A(2), B(2)), \\
 & D_{1,2} + d(A(2), B(2)), \\
 & D_{2,1} + d(A(2), B(2)).
 \end{aligned}$$

Da die Zuordnung mit dem möglichst kleinsten summierten Abstand gesucht wird, ergibt sich der Gesamtabstand bis Schritt zwei zu

$$D_{2,2} = \min(D_{1,1}, D_{1,2}, D_{2,1}) + d(A(2), B(2)).$$

Aus der Verallgemeinerung dieser Formel ergibt sich für den Gesamtabstand die Rekursionsgleichung

$$D_{p,q} = \min(D_{p-1,q-1}, D_{p-1,q}, D_{p,q-1}) + d(A(p), B(q)).$$

Der aufsummierte Abstand $D_{DTW}(A, B)$ ergibt sich beim Erreichen des Endpunktes in (P, P) aus

$$D_{DTW}(A, B) = D_{P,P}.$$

Der Abstand der optimalen Zuordnung ist aus der Tabelle somit im oberen rechten Feld abzulesen. Wird zusätzlich in jedem Schritt in der Tabelle durch einen Pfeil vermerkt, welches Vorgängerfeld durch das Bestimmen des minimalen Abstandswertes ausgewählt wurde, so kann die optimale Zuordnung ebenfalls aus der Tabelle ermittelt werden. Ausgehend vom Endpunkt kann der Weg durch die Tabelle über

die Pfeile konstruiert werden. Dieser Weg zeigt den Graph der Zuordnungsfunktion, auch als Zuordnungspfad bezeichnet. Abbildung 17.5(b) zeigt anhand eines Beispiels zweier fiktiver Sprachmuster, wie sich der Zuordnungspfad (rot markiert) aus den ermittelten Pfeilen ergibt, wenn der Weg ausgehend vom letzten Zustand oben rechts zum Startzustand entgegen der Pfeilrichtungen zurückverfolgt wird. In diesem Beispiel ergibt sich ein eindeutiger Zuordnungspfad. Im Allgemeinen ist die Lösung des Optimierungsproblems allerdings nicht eindeutig, und es kann mehrere Zuordnungspfade geben, welche die Muster optimal aufeinander anpassen. Bei identischen Sprachmustern ergibt sich die Diagonale als Zuordnungspfad. Je weniger die Sprachmuster aufeinander angepasst werden müssen, desto weniger weicht der Zuordnungspfad von der Diagonalen ab.

Die Klassifizierung erfolgt anschließend mit den angepassten Sprachmustern. Bei jeder Abfrage wird das Testmuster T_j mithilfe der Dynamik Time Warping Methode an jedes Referenzmuster zeitlich angepasst und gleichzeitig der Gesamtabstand der angepassten Sprachmuster ermittelt. Mit diesem ermittelten Gesamtabstand pro Referenzmuster kann anschließend das Nächster-Nachbar-Verfahren ausgeführt werden.

Kernaspekte: Zentral in diesem Abschnitt ist somit der Vergleich von Zeitreihen. Erneut spielt das mathematische Konzept eines Abstandsmaßes eine wichtige Rolle, allerdings sei nochmal darauf hingewiesen, dass im mathematischen Sinne keine DTW-Distanz existiert. Die dynamische Zeitanpassung kann allerdings als das Minimieren der euklidischen Distanz aufgefasst werden. Das Optimierungsproblem kann mit dynamischer Programmierung gelöst werden.

17.4 Bewertung des Spracherkennungssystems

Nachdem in der *Trainingsphase* anhand der Trainingsdaten ein Verfahren zur Klassifikation der Sprachsignale ermittelt wurde, wird in der *Testphase* die Performance des Spracherkennungssystems evaluiert, bevor das Spracherkennungssystem in der *Anwendungsphase* tatsächlich für das Erkennen von Sprachsignalen in der Praxis verwendet werden kann. Konkret soll in der Testphase untersucht werden, wie gut die Klassenzugehörigkeit für Datenpunkte, die noch nicht in der Trainingsphase verwendet wurden, vorhergesagt wird. Für die Entwicklung und Evaluation eines Modells wird somit ein ausreichend großer gelabelter Datensatz benötigt, der in (mindestens) zwei Datensätze aufgeteilt wird: Die Trainingsdaten und die Testdaten. Die Trainingsdaten sind bereits in Abschnitt 17.2 als Tupel (R_i, W^{R_i}) , $i = 1, \dots, r$, mit den Referenzmustern R_i und der zugehörigen Wortbedeutung W^{R_i} mit $W^{R_i} \in \{1, \dots, K\}$ definiert. Als Testdaten wird entsprechend das Tupel (T_j, W^{T_j}) , $j = 1, \dots, t$, mit den ermittelten Testmustern T_j und der zugehörigen Wortbedeutung W^{T_j} mit $W^{T_j} \in \{1, \dots, K\}$ bezeichnet.

Im Falle der Einzelworterkennung kann die Bewertung des Spracherkennungssystems über die für Klassifikationsverfahren bekannten Qualitätsmaße erfolgen (Jurafsky & Martin, 2023, S. 344). Hierzu wird in der Testphase die Klassifikation für die Testdaten, deren Klassenzugehörigkeit bereits bekannt sind, mehrmals durchgeführt. Die

Tabelle 17.1: Wahrheitsmatrix für eine binäre Klassifikation mit n Testdatenpunkten

		klassifiziertes Klassenlabel		Summe
		positiv klassifiziert	negativ klassifiziert	
tatsächliches Klassenlabel	tatsächlich positiv	RP (richtig positiv)	FN (falsch negativ)	$RP + FN$
	tatsächlich negativ	FP (falsch positiv)	RN (richtig negativ)	$FP + RN$
Summe		$RP + FP$	$FN + RN$	n

Klassifikationsergebnisse können bewertet werden, indem die vorhergesagte Klassenzugehörigkeit mit der tatsächlichen Klassenzugehörigkeit verglichen wird. Jedes klassifizierte Beispiel kann als Tupel $(W_R^{T_j}, W^{T_j})$, $j = 1, \dots, t$, dargestellt werden. Die verschiedenen Klassenzugehörigkeiten der Testdaten können in Form einer Wahrheitsmatrix – auch unter dem Namen Konfusionsmatrix (engl. Confusion Matrix) bekannt – dargestellt werden². Die Zeilen der Matrix stellen den tatsächlichen Wert des Klassenzugehörigkeit W^{T_j} dar, während die Spalten den vorhergesagten Wert $W_R^{T_j}$ zeigen. Tabelle 17.1 zeigt die Wahrheitsmatrix beispielhaft für eine binäre Klassifikation mit den Klassen „positiv“ und „negativ“. Die Anzahl der korrekt klassifizierten Datenpunkte (richtig positiv und richtig negativ) – getrennt nach Klassen – findet sich auf der Hauptdiagonale. Die Anzahl der falsch klassifizierten Beispiele (falsch positiv und falsch negativ) befindet sich auf der Gegendiagonale. Aus der Wahrheitsmatrix lassen sich verschiedene Qualitätsmaße ableiten, von denen im Folgenden die verbreitetsten und zugänglichsten vorgestellt werden.

Die *Genauigkeit* (engl. *accuracy*) zählt als das verbreitetste Qualitätsmaß (Sokolova & Lapalme, 2009, S. 427 - 428). Sie gibt das Verhältnis der richtig klassifizierten Datenpunkte zu allen Datenpunkten an (Heesen, 2023, S. 297). Sie lässt sich mit der Gleichung

$$\text{accuracy} := \frac{\text{Anzahl der richtig klassifizierten Datenpunkte}}{\text{Anzahl aller Datenpunkte}}$$

²Die Wahrheitsmatrix erinnert an die Wirkungstabelle eines Hypothesentests. Wird ein binäres Klassifikationsproblem betrachtet, so lässt sich dieses zwar als statistisches Testproblem interpretieren, die Fehler erster und zweiter Art werden hier allerdings nicht getrennt voneinander sondern aufsummiert berücksichtigt. Zudem liegt eine andere Zielsetzung vor: Beim Hypothesentest wird eine Hypothese gegen eine Alternative, getestet und es ist ein festgelegtes Niveau einzuhalten. Bei einem Klassifikationsproblem werden alle Fehlentscheidungen als gleich schwerwiegend angesehen, und es wird im Vorhinein keine Schranke für den Fehler erster Art festgelegt (Trabs et al., 2021, S. 187).

berechnen. Für die in Tabelle 17.1 abgebildete Wahrheitsmatrix ergibt sich die Genauigkeit zu

$$\text{accuracy} = \frac{RP + RN}{n}.$$

Die Genauigkeit eignet sich weniger gut als Qualitätsmaß, wenn der Testdatensatz nicht ausgewogen ist, die negative und die positive Klasse somit eine sehr unterschiedliche Anzahl von Testdatenpunkten enthalten (Kubat, Matwin et al., 1997). Gibt es sehr viel mehr Testdatenpunkte in der positiven Klasse und der Klassifikator ordnet alle Datenpunkte kategorisch der positiven Klasse zu, so ist die Genauigkeit groß, obwohl dieser Klassifikator nutzlos ist. Die Verwendung der Genauigkeit als Maß für die Qualität des Klassifikators kann in solchen Szenarien zu einer irreführenden Interpretation der Ergebnisse führen. Dieser Effekt wird bei der Vorstellung des Lernmaterials am Beispiel eines fiktiven Spracherkennungssystems, welches alle Sprachsignale einer einzigen Klasse zuordnet, genauer erläutert (vgl. Kapitel 19.2.5).

Es lohnt sich daher, weitere Qualitätsmaße wie die *Präzision* zu betrachten. Für die Präzision wird im Gegensatz zur Genauigkeit jede Klasse einzeln betrachtet. Sie gibt den Anteil der richtig als Klasse x klassifizierten Datenpunkte von allen als Klasse x vorhergesagten Datenpunkten an (Botsch, 2023, S. 13):

$$\text{precision}_x := \frac{\text{Anzahl der richtig als } x \text{ klassifizierten Datenpunkte}}{\text{Anzahl aller als } x \text{ klassifizierten Datenpunkte}}.$$

Die Präzision für die Klasse mit dem Label „positiv“ ergibt sich somit zu

$$\begin{aligned} \text{precision}_{\text{pos}} &= \frac{\text{Anzahl der richtig als positiv klassifizierten Datenpunkte}}{\text{Anzahl aller als positiv klassifizierten Datenpunkte}} \\ &= \frac{RP}{RP + FP} \end{aligned}$$

und für die Klasse mit der Klassenzugehörigkeit „negativ“ zu

$$\begin{aligned} \text{precision}_{\text{neg}} &= \frac{\text{Anzahl der richtig als negativ klassifizierten Datenpunkte}}{\text{Anzahl aller als negativ klassifizierten Datenpunkte}} \\ &= \frac{RN}{FN + RN}. \end{aligned}$$

Bei einem guten Klassifikator sind die Genauigkeit und die Präzision der einzelnen Klassen hoch.

Im Falle der Spracherkennung stellen die Datenpunkte die Sprachmuster und die Klassenzugehörigkeit die Wortbedeutung dar. In der Testphase wird mit dem Spracherkennungssystem systematisch für mehrere Sprachsignale die Wortbedeutung ermittelt sowie deren Korrektheit überprüft und festgehalten.

Kernaspekte: Die wesentlichen mathematischen Konzepte zur Bewertung von Klassifizierungssystemen wie dem vorgestellten Spracherkennungssystem basieren auf der Konstruktion von Mehrfeldertafeln (Wahrheitsmatrizen) zur Erfassung der absoluten

Häufigkeiten der Klassifikationsergebnisse. Aus diesen lassen sich durch die Bildung relativer Häufigkeiten die gängigen Qualitätsmaße wie beispielsweise die Genauigkeit und Präzision ableiten.

18 Anwendung der mathematischen Methoden mit realen Sprachsignalen

In dieser Arbeit wird ein Datensatz von Warden (2018) verwendet, der Audiosignale von 35 verschiedenen englischsprachigen Wörtern im WAV-Format enthält. Für die Entwicklung des Lernmaterials werden die Sprachsignale der englischen Zahlenwörter eins bis vier, die von verschiedenen Sprechern stammen, ausgewählt. Insgesamt werden 240 Sprachsignale verwendet, davon sind zu gleichen Teilen die Sprachsignale mit der Wortbedeutung „one“ und „two“, „three“ und „four“ enthalten. 80 % der Daten werden für das Training des Modells und 20 % für die Bewertung genutzt. Da die Aufteilung in Trainings- und Testdaten jedoch zufällig erfolgt, ist nach der Teilung möglicherweise keine gleichmäßige Verteilung der Wortklassen in den beiden Datensätzen gewährleistet.

Die Klassifizierung der Testdaten erfolgt gemäß der in Abbildung 17.1 dargestellten Schritte, die in Kapitel 17 detailliert erläutert sind. Die Fourier-Transformation wird aus Effizienzgründen (Mallat, 2009, S. 78) jedoch mithilfe der schnellen Fourier-Transformation (engl. Fast Fourier-Transformation, FFT) umgesetzt. Als Fensterfunktion wird die in Abschnitt 17.1.2 vorgestellte Hamming-Funktion verwendet. Das anschließende Extrahieren der Merkmale erfolgt mit der Mel-Filterung. Eine in der Spracherkennung typische Anzahl an Filtern liegt zwischen 20 und 30 (Euler, 2006, S. 36). Die Anzahl der Mel-Filter wird hier deshalb auf 25 festgelegt. Die Klassifizierung erfolgt mit dem in Abschnitt 17.2.3 beschriebenen Nächster-Nachbar-Verfahren mit der euklidischen Metrik als Abstandsmaß. Zusätzlich wird als Modellverbesserung die Anpassung der Sprachmuster über das Dynamic Time Warping mit den in Abschnitt 17.3.1 vorgestellten symmetrischen Randbedingungen durchgeführt.

Für die Umsetzung der mathematischen Methoden im Code wird teilweise auf bereits entwickelte Pakete verschiedener Bibliotheken zurückgegriffen. Neben den Standardbibliotheken sind hier besonders die Bibliotheken „Librosa“ (McFee et al., 2024) und „dtw“ (Giorgino, 2009) zu nennen, die durch eine performante Architektur eine effiziente Berechnung des Mel-Spekrogramms und des Dynamic Time Warpings des Spracherkennungssystems ermöglichen. Diese Bibliotheken tragen entscheidend dazu bei, dass die Erkennung von Sprachsignalen innerhalb weniger Sekunden erfolgen kann und den Lernenden ein praxisnaher Einsatz des Spracherkennungssystems vermittelt wird. Um jedoch die zugrunde liegenden Methoden transparent darzustellen, werden auf den Arbeitsblättern die Methoden zunächst ohne den direkten Einsatz

		Wahrheitstabelle			
		one	two	three	four
tatsächliches Wort	one	6	2	0	3
	two	2	6	3	2
	three	0	2	5	1
	four	1	1	1	13
		one	two	three	four
		erkanntes Wort			

		Wahrheitstabelle			
		one	two	three	four
tatsächliches Wort	one	9	2	0	0
	two	0	11	1	1
	three	0	2	5	1
	four	0	0	0	16
		one	two	three	four
		erkanntes Wort			

(a) Wahrheitstabelle des Spracherkennungssystems mit einfacherem Mustervergleich

(b) Wahrheitstabellen des Spracherkennungssystems mit DTW-Verfahren

Abbildung 18.1: Wahrheitstabelle der entwickelten Spracherkennungssysteme

der Bibliotheken implementiert. Dies ist besonders wichtig, da viele Funktionen der Bibliotheken mehrere Methoden in einem einzigen Aufruf zusammenfassen, wie etwa die Ermittlung des Mel-Spektrogramms und die Funktionsweise so nicht sinnvoll untersucht werden kann. Erst bei der Erprobung des Spracherkennungssystems wird auf implementierten Funktionen der Bibliotheken zurückgegriffen.

Abbildung 18.1 zeigt die Klassifikationsergebnisse des Spracherkennungssystems, welches mit einfacherem Mustervergleich funktioniert, und des Spracherkennungssystems, das mit dem Verfahren des Dynamic Time Warping und anschließendem Mustervergleich arbeitet. Bereits anhand der Wahrheitstabellen (vgl. Abbildung 18.1) fällt auf, dass das Spracherkennungssystem mit DTW mehr Sprachsignale korrekt klassifiziert hat als das Spracherkennungssystem ohne DTW. Die Berücksichtigung der zeitlichen Verschiebungen der Sprachmuster scheint somit zu der erwarteten Verbesserung der Performance zu führen.

Die Bewertung der Spracherkennungssysteme erfolgt anschließend mit den in Abschnitt 17.4 kennengelernten Qualitätsmaßen. Tabelle 18.1 zeigt die ermittelten Werte für die Präzision der einzelnen Klassen und die Genauigkeit des Spracherkennungssystems mit DTW und des Spracherkennungssystems ohne DTW. Der Wert der Präzision jeder Klasse sowie der Wert der Genauigkeit liegen für das Spracherkennungssystem mit DTW im Vergleich zum Spracherkennungssystem ohne DTW näher bei eins. Die Qualitätsmaße bestätigen somit die Vermutung, dass das Spracherkennungssystem durch die zeitliche Anpassung der Sprachmuster verbessert wird. Mit einer Genauigkeit von 0,85 liefert das Spracherkennungssystem eine solide Leis-

Tabelle 18.1: Qualitätsmaße verschiedener Spracherkennungssysteme

Sprach- erkennungs- system	Genauigkeit	Präzision Klasse „one“	Präzision Klasse „two“	Präzision Klasse „three“	Präzision Klasse „four“
ohne DTW	0,62	0,67	0,55	0,56	0,68
mit DTW	0,85	1	0,73	0,83	0,89

tung, die zwar nicht ganz an die Spitzenwerte aus der Forschung heranreicht¹, aber für Bildungszwecke absolut zufriedenstellend ist.

¹C. S. Myers et al. (1980) erreichten bei der Einzelworterkennung mit DTW je nach Randbedingungen Fehlerraten von 0,04 bis 0,06, was einer Genauigkeit von 0,94 bis 0,96 entspricht. Auch Jiang und Chen (2023) ermitteln eine Genauigkeit von 0,94 bei der Einzelworterkennung mit DTW.

19 Umsetzung im Lernmaterial

In diesem Kapitel wird gezeigt, wie mit Lernenden ein KI-System zur Erkennung gesprochener Sprache unter Einbezug bekannter Methoden aus der Schulmathematik entwickelt werden kann. Hierzu setzen sie sich mit der mathematischen Darstellung von einzelnen Tönen und komplexeren Signalen auseinander, lernen Methoden für die Extraktion von Merkmalen aus einem Signal und für die Klassifikation dieser Merkmale kennen. Anschließend evaluieren sie ihr Spracherkennungssystem auf einem Datensatz mit den Zahlenwörtern „one“, „two“, „three“ und „four“ und erarbeiten das Dynamic Time Warping als eine mögliche Modellverbesserung. Eine Übersicht über den Aufbau und die Inhalte des Lernmaterials im Rahmen eines Modellierungstages findet sich in Tabelle 19.1.

Für die Entwicklung des Lernmaterials werden zu den in Kapitel 17 vorgestellten mathematischen Inhalten kleinschrittige Aufgaben formuliert und diese zu einer kohärenten Lernumgebung zusammengeführt. Da die didaktische Reduktion auf den in Kapitel 17 identifizierten Inhalten basiert, werden die bereits dort aufgeführten Quellenverweise hier nicht erneut genannt.

Das Material ist für Lernende ab Jahrgangsstufe 10 konzipiert. Vor dessen Einsatz sollte überprüft werden, ob das in Abschnitt 19.1 beschriebene Vorwissen bereits vorhanden ist. Da einige der vorausgesetzten Inhalten je nach Bundesland und dortigen curricularen Vorgaben erst in Jahrgangsstufe 10 eingeführt werden, ist eine vorherige Abstimmung mit dem Vorwissen empfehlenswert.

Zum Lernmaterial gehören digitale Arbeitsblätter, ergänzende Tipp-, Informations-Karten und Zusatzblätter, sowie begleitende Präsentationsfolien. Bildschirmfotos des Lernmaterials sowie das zugehörige Begleitmaterial sind in Anhang B zu finden. In den nachfolgenden Abschnitten wird an den passenden Stellen erneut auf die einzelnen im Anhang abgedruckten Elemente des Lernmaterials verwiesen. Die Lesenden seien zudem eingeladen, die Lernplattform¹ zu besuchen, auf der das Lehr- und Lernmaterial in seiner vollen Funktion eingesehen werden kann.

¹Der Zugriff zur Lernplattform (workshops.cammp.online), die technischen Umsetzung sowie der grundsätzliche Aufbau der Materialien werden ausführlich in Kapitel 8 beschrieben.

Tabelle 19.1: Tabellarische Übersicht über das Lernmaterial zur Spracherkennung (Dauer in Minuten)

Phase	Inhalt	Dauer
Einstieg	Basiswissen zu Spracherkennung und KI, Einführung in die mathematische Darstellung von Tönen, Vereinfachung und erste Ideen zu Übersetzung der Realsituation in ein mathematisches Modell	20
Technik-Einführung		10
Arbeitsblatt 1	Untersuchung verschiedener Töne als periodische Sinusschwingungen und deren Funktionsgleichung	40
Besprechung 1	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 1, Vorstellen der Fourieranalyse als Methode zur Extraktion einzelner Teiltöne eines Tongemisches	5
Arbeitsblatt 2	Erstellung von Dreiklängen, Erkennung von Dreiklängen, Nachbildung eines Dreiklangs	40
Besprechung 2	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 2, Annahmen abändern: Von Untersuchung von Tönen zu Untersuchung von Sprachsignalen	5
Arbeitsblatt 3	Ermittelte Frequenzen zu Merkmalen zusammenfassen	35
Besprechung 3	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 3, Ideen für die Abstandsbestimmung sammeln	10
Arbeitsblatt 4	Klassifizierung der Sprachmuster	20
Besprechung 4	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 4, Ideen für die Bewertung sammeln	5
Arbeitsblatt 5	Bewertung des Spracherkennungssystems	20
Besprechung 5	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 5, Diskussion der Bewertungsergebnisse, Ideen für Modellverbesserungen sammeln	10
Arbeitsblatt 6	Dynamische Zeitanpassung der Sprachmuster	60
Besprechung 6 mit Abschluss	Zusammenfassung der Vorgehensweise auf Arbeitsblatt 6, Sammlung von Problemen und möglichen Anwendungsbereichen des Spracherkennungssystems, Diskussion zu ethisch bedenklichen/ problematischen Aspekten bei der Nutzung und Entwicklung eines Spracherkennungssystems	10

19.1 Vorwissen und Lernziele

Im Lernmaterial sollen die wichtigsten Schritte für die Entwicklung und Evaluierung eines Spracherkennungssystems durchlaufen und eine mögliche mathematische Methode für deren Umsetzung kennengelernt werden. Zentral ist die Aufbereitung der Daten einer Sprachaufnahme, die Klassifizierung der Sprachaufnahme und schließlich die Bewertung des Klassifizierungsverfahrens. Dabei ist es notwendig, dass mathematische Inhalte, die teilweise über das Schulniveau hinaus gehen, erarbeitet und anschließend problembezogen angewandt werden. Zugleich soll die mathematische Modellierung als Metakonzept für eine mathematische Herangehensweise an reale Probleme kennengelernt werden. Konkret sollen die folgenden **Lernziele** erreicht werden:

Die Lernenden können...

- ... die Sinusfunktion als eine mögliche mathematische Repräsentation eines (reinen) Tons nennen und die physikalischen Eigenschaften eines Tons wie Lautstärke und Tonhöhe den mathematischen Parametern Amplitude und Frequenz zuordnen, sowie die Auswirkung der Änderung dieser Parameter auf den Graphen der Sinusfunktion beschreiben. (vgl. Tabelle 6.1 L1)
- ... die Sinusfunktion als Formel so anpassen, dass die Frequenz bzw. die Amplitude linear von der Zeit abhängt. (vgl. Tabelle 6.1 L1)
- ... die Grundidee der Fouriertransformation erläutern. (vgl. Tabelle 6.1 L2)
- ... die Fouriertransformation im Kontext der akustischen Vorverarbeitung anwenden und deren Ergebnis interpretieren. (vgl. Tabelle 6.1 L2)
- ... die Notwendigkeit einer Fensterung bei der Fourieranalyse von Signalen mit zeitlich variabler Frequenzzusammensetzung erläutern. (vgl. Tabelle 6.1 L2)
- ... verschiedene Methoden (besonders hohe und tiefe Werte streichen, Werte per Mittelwertbildung zusammenfassen, Werte unterschiedlich stark gewichten) zur Reduktion von Merkmalen nennen. (vgl. Tabelle 6.1 L2)
- ... unterschiedliche statistische Kenngrößen einer Datenmenge hinsichtlich deren Eignung für die Zusammenfassung einzelner Frequenzwerte vergleichen. (vgl. Tabelle 6.1 L2)
- ... den euklidischen Abstand zwischen zwei Vektoren im mehrdimensionalen Raum bestimmen und seinen Wert als Maß für die Ähnlichkeit der Vektoren interpretieren. (vgl. Tabelle 6.1 L3)
- ... das Nächster-Nachbar-Verfahren erläutern und anwenden. (vgl. Tabelle 6.1 L5)
- ... die Rolle der (Trainings- und Test-) Daten bei der Entwicklung und Evaluation von ML-Methoden am Beispiel eines Spracherkennungssystems erläutern. (vgl. Tabelle 6.1 L4, L8)
- ... eine Wahrheitstabelle lesen und die eingetragenen Werte im Anwendungskontext interpretieren, sowie eine Wahrheitstabelle aus gegebenen Klassifikationsergebnissen erstellen. (vgl. Tabelle 6.1 L8, L9)

- ... Qualitätsmaße wie die Genauigkeit und die Präzision bestimmen, sowie deren Bedeutung für die Qualität eines Klassifikationsverfahrens interpretieren. (vgl. Tabelle 6.1 L10, L11)
- ... Überlegungen zur Verbesserung der Modellierung anstellen. (vgl. Tabelle 6.1 L7)
- ... die Idee des Dynamic Time Warpings am Graphen erklären. (vgl. Tabelle 6.1 L5)
- ... Eigenschaften von natürlicher Sprache als mathematische Randbedingungen der Zuordnungsfunktion formulieren. (vgl. Tabelle 6.1 L1, L3)
- ... das Verfahren des Dynamic Time Warpings im eindimensionalen Fall durchführen. (vgl. Tabelle 6.1 L5)
- ... Probleme und Grenzen des entwickelten Modells benennen. (vgl. Tabelle 6.1 L11)
- ... zentrale ethische Aspekte hinsichtlich Entwicklung und Einsatz von Spracherkennungssystemen nennen und beurteilen. (vgl. Tabelle 6.1 L12)

Um eine erfolgreiche Bearbeitung des Lernmaterials zu ermöglichen, wird Vorwissen zu verschiedenen mathematischen Inhalten vorausgesetzt. Eine detaillierte Einordnung dieser Inhalte sowie die Zuordnung zu den entsprechenden Jahrgangsstufen erfolgt im Abschnitt zur curricularen Anbindung (vgl. Abschnitt 13.3). Besonders relevant sind dabei Kenntnisse zu

- Linearen, quadratischen und trigonometrischen Funktionen,
- Lagemaßen, insbesondere dem Mittelwert,
- Abstand zweier Punkte im zweidimensionalen Raum,
- relativer Häufigkeit und Vierfeldertafeln.

19.2 Erläuterung des Lehr- und Lernmaterials

Im Folgenden wird der Aufbau des entwickelten Lernmaterials skizziert und didaktisch erläutert. Das Lernmaterial orientiert sich in Teilen an den Arbeiten von Hoeffer (2022) und Steffen (2016). Erstere beschäftigt sich mit der Entwicklung von computergestütztem Lernmaterial zur Aktivitätserkennung. Die Evaluation des Klassifizierungsverfahrens, insbesondere Aufgabe 2 und 3 auf dem Arbeitsblatt 5 orientieren sich an dieser Arbeit.

In Steffen (2016) wurde die Musikerkennung am Beispiel der App Shazam für Lernende der Mittelstufe im Rahmen eines Modellierungstages erarbeitet und erprobt. Während bei der Musikerkennung zur Identifikation eines Songs eine eindeutige Abfolge verschiedener Merkmale für jeden Song herangezogen werden kann, ist es bei der Spracherkennung nicht möglich, eine eindeutige Folge von Erkennungsmerkmalen für jedes Wort zu bestimmen. Sprache ist sehr variabel, und damit sind auch die

ermittelten Merkmale für ein gesprochenes Wort nicht immer gleich. Ein Abgleich der ermittelten Merkmalsabfolge mit einer Datenbank, wie es bei der Musikerkennung üblich ist, genügt daher nicht. Die Ausgangssituation ist jedoch ähnlich: Es liegen Audiosignale vor, deren enthaltenen Frequenzen mithilfe der Fouriertransformation ermittelt werden können. Die im Rahmen dieser Arbeit hinzugefügten oder modifizierten Inhalte beziehen sich primär auf die Arbeitsblätter 3 bis 6. Daher wird deren Umsetzung im Folgenden besonders detailliert erläutert. Arbeitsblatt 1 und Arbeitsblatt 2 werden an den neuen Anwendungskontext angepasst, sowie das Design aktualisiert und durch einfach zu bedienende Programmelemente wie Drop-Down-Fenster und ähnliches interaktiver gestaltet. Auch wenn inhaltlich wenig Änderungen stattgefunden haben, so werden die auf Arbeitsblatt 1 und 2 behandelten Inhalte jedoch ebenfalls kurz skizziert, um das Verständnis zu gewährleisten.

19.2.1 Arbeitsblatt 1: Einen Ton mathematisch modellieren

In einer Einstiegspräsentation (vgl. Anhang B.7) wird zunächst eine Einführung in das Thema Spracherkennung gegeben. Anwendungen, bei welchen die Spracherkennung verwendet wird, werden gesammelt, um die Lernenden in ihrer Lebenswelt abzuholen. Anschließend wird die Verbindung zwischen KI und Spracherkennung erarbeitet, indem zunächst das Vorwissen der Lernenden zu KI und deren Anwendungsbereichen aktiviert wird. Da die gesprochene Sprache als natürliche Form der Mensch-Maschine-Interaktion gilt, wird die Spracherkennung als wesentlicher Bestandteil vieler KI-Systeme und somit als zentrale Aufgabe dieses Forschungsbereichs herausgestellt. Nach einem Überblick über wichtige Meilensteine in der Entwicklung von KI-Systemen mit einem Fokus auf Systeme, die Sprache erkennen, wird das grundlegende Verfahren der Spracherkennung wie folgt veranschaulicht: Sprachsignale mit unbekannter Wortbedeutung werden mit Signalen verglichen, deren Wortbedeutung bekannt ist.

Ⓜ Einordnung in den Modellierungskreislauf

Eine kurze Einführung des Modellierungskreislaufs erfolgt, wobei das reale Problem als das Erkennen von gesprochener Sprache identifiziert wird (vgl. Anhang B.7). Dieses Problem wird zunächst vereinfacht, indem statt eines komplexen Sprachsignals nur ein einzelner (reiner) Ton erkannt werden soll. Erst später werden komplexere Signale mit zeitlich variabler Frequenzzusammensetzung, wie es bei Sprachsignalen der Fall ist, betrachtet. So kann der Modellierungsprozess in mehrere Durchläufe unterteilt werden, um die Komplexität zu reduzieren. Der Schritt der Mathematisierung erfolgt auf dem ersten Arbeitsblatt. Der Modellierungskreislauf dient während des gesamten Modellierungsworkshop als Orientierung. Immer wieder werden in den Plenumsphasen die auf den Arbeitsblättern erarbeiteten Teilschritte im Kreislauf eingesortiert und die Inhalte so in den Gesamtprozess eingeordnet. Der Modellierungskreislauf wird somit, wie von Schukajlow et al. (2011, S. 42 - 43), vorgeschlagen, als metakognitives, visuelles Element genutzt, das die Reflexion des Modellierungsprozesses unterstützen und den Aufbau von Metawissen zur mathematischen Modellierung fördern soll. Auch in dieser Erläuterung des Lernmaterials erfolgt an mehreren

Stellen eine Einordnung in den Modellierungskreislauf, die für eine bessere Sichtbarkeit mit dem Symbol des umkreisten Buchstabens „M“ versehen wird.

Auf dem ersten Arbeitsblatt (vgl. Anhang B.1) wird die mathematische Modellierung von Tönen thematisiert. Dabei wird das wohl einfachste Geräusch, der (reine) Ton behandelt, dessen erzeugende Schwingung durch eine Sinusfunktion

$$g(t) = A \cdot \sin(2\pi ft)$$

mit Amplitude A und Frequenz f beschrieben werden kann. t steht dabei für die Zeit.

Die Lernenden befassen sich mit dem Zusammenhang der Parameter A und f mit den charakteristischen physikalischen Eigenschaften eines Tons, der Laustärke und der Tonhöhe. Dazu erzeugen sie durch Anpassen des Parameters A einen zu einem gegebenen Grundton mit $A = 1$ und $f = 200$ Hz lauteren und leiseren Ton. Außerdem ergänzen sie eine Zeitabhängigkeit der Amplitude, um zu bewirken, dass die Lautstärke im Laufe der Zeit zu- oder abnimmt. Zudem wird der Zusammenhang von Tonhöhe und Frequenz untersucht. Hierzu ist es notwendig, die Frequenz so anzupassen, dass ein Ton mit einer höheren beziehungsweise einer tieferen Tonhöhe erzeugt wird. Auch hier soll anschließend wieder eine Zeitabhängigkeit der Frequenz hinzugefügt werden, sodass der Ton zum Beispiel zunächst höher und anschließend tiefer wird.

Die Lernenden erhalten als Ausgabe sowohl den Graphen ihrer eingegebenen Sinusfunktion als auch die zugehörige Audiodatei, die im digitalen Arbeitsblatt abgespielt werden kann. Dadurch haben sie die Möglichkeit, ihre Ergebnisse eigenständig sowohl visuell als auch auditiv zu überprüfen. Wann immer es in anderen Aufgabenteilen möglich ist, wird diese visuelle und auditive Ausgabeform beibehalten, um die Selbsttätigkeit sowie die Kompetenz zur Validierung zu fördern.

19.2.2 Arbeitsblatt 2: Die Fouriertransformation am Beispiel von Dreiklängen

Ⓜ Einordnung in den Modellierungskreislauf

Nachdem auf dem ersten Arbeitsblatt die mathematische Beschreibung von reinen Tönen untersucht wurde, sollen nun komplexere Audiosignale in Form von Dreiklängen in den Blick genommen werden. Das vereinfachte Problem kann somit als das Erkennen von Dreiklängen formuliert werden.

Ein Dreiklang ergibt sich als Überlagerung dreier Sinusschwingungen mit einem bestimmten Abstand der Tonhöhen zu

$$s(t) = \sin(2\pi f_a t) + \sin(2\pi f_b t) + \sin(2\pi f_c t).$$

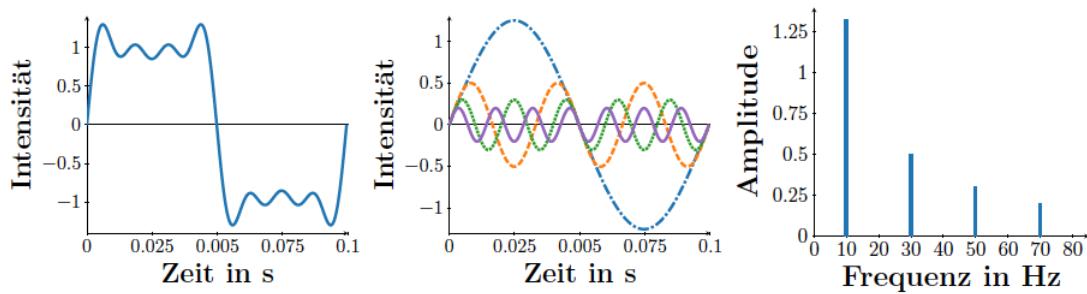


Abbildung 19.1: Funktionsweise der Fouriertransformation ausgehend von links Ausgangssignal, mittig Zerlegung und rechts Amplitudenspektrum

Um auch hier die Frequenzen als charakteristische Eigenschaft des Audiosignals ermitteln zu können, genügt das Ablesen am Graphen nicht mehr. Zu dieser Erkenntnis gelangen die Lernenden auf Arbeitsblatt 2 (vgl. Anhang B.2). Anschließend wird deshalb die Grundidee der Fourier-Analyse dargelegt. Dabei wird ausgenutzt, dass jedes akustische Signal als Summe von Sinusschwingungen verschiedener Frequenzen dargestellt werden kann. Bei der Fouriertransformation wird das akustische Ausgangssignal in seine einzelnen Schwingungen zerlegt und die Frequenzen der Sinusschwingungen, sowie deren zugehörigen Amplituden ermittelt. Die Funktionsweise der Fourier-Transformation wird im Lernmaterial anhand Abbildung 19.1 veranschaulicht. Mit der Fouriertransformation wird anschließend für mehrere Audiosignale das Amplitudenspektrum ermittelt.

Die eigenständige Erstellung des Spektrums ist allerdings nicht trivial und würde eine tiefergehende Einführung in die Signalverarbeitung erfordern. Andererseits folgen noch viele weitere Modellierungsschritte, weshalb die Funktionsweise der Fouriertransformation hier nur anschaulich gezeigt, die Berechnung des Amplitudenspektrums aber in einer zeitlich kürzeren Variante als Grey Box durchgeführt wird. So erfolgt keine Überforderung, es bleibt mehr Zeit für weitere Modellierungsschritte, und der Fokus kann auf die Interpretation des Amplitudenspektrums gelegt werden. Dazu erhalten alle die Aufgabe, die Frequenzen im Amplitudenspektrum verschiedener Dreiklänge abzulesen und den zugehörigen Dreiklang zu ermitteln.

19.2.3 Arbeitsblatt 3: Merkmale reduzieren

④ Einordnung in den Modellierungskreislauf

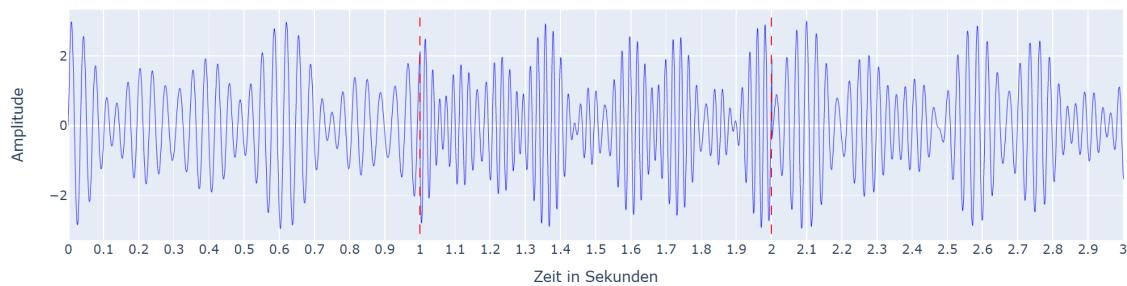
Nachdem die Schülerinnen und Schüler auf dem zweiten Arbeitsblatt kennengelernt haben, wie die Frequenzen der in einem Dreiklang enthaltenen Sinusschwingungen ermittelt werden können, und der Modellierungskreislauf ein erstes Mal durchlaufen wurde, wird auf Arbeitsblatt 3 die gemachte Vereinfachung modifiziert. So kann sich schrittweise an das eigentliche Problem angenähert werden. Das vereinfachte Problem soll nun sein, komplexere akustische Signale, wie einzelne Wörter zu erkennen. Insbesondere soll eine zeitliche Variation der Frequenzzusammensetzung in den Signalen zugelassen werden.

Das erste Ziel des dritten Arbeitsblatts (vgl. Anhang B.3) besteht darin, zu erarbeiten, wie die Frequenzen von Signalen mit zeitlich variabler Frequenzzusammensetzung ermittelt werden können. Es wird daher die gefensterte diskrete Fouriertransformation behandelt. Im Lernmaterial wird das in Abschnitt 17.1.2 eingeführte Hammingfenster verwendet, allerdings wird die Fensterfunktion mit den Lernenden nicht thematisiert, sondern lediglich auf die Tatsache eingegangen, dass die Fouriertransformation für einzelne Zeitblöcke separat durchgeführt wird. Anschließend sollen die Schülerinnen und Schüler Verfahren kennenzulernen, mit denen die ermittelten Frequenzen geeignet reduziert werden können. Dies ist wichtig, um die Sprachsignale später über eine kleine Anzahl an Merkmalen charakterisieren zu können. Im Lernmaterial wird die Mel-Filterung auf die Sprachsignale angewandt. Der Prozess der Mel-Filterung kann theoretisch als Matrix-Vektor-Multiplikation mit dem Vektor der Fouriertransformierten im p -ten Fenster dargestellt werden (vgl. Abschnitt 17.1.3). Allerdings weist die resultierende Matrix aufgrund der nicht äquidistanten Abstände zwischen den Frequenzbändern eine unregelmäßige Struktur auf. Für die Schülerinnen und Schüler ist die Mel-Filterbank als Matrix daher sehr abstrakt. Mit den Lernenden wird die Mel-Filterung deshalb als Mittelwertbildung mit unterschiedlich breiten Frequenzblöcken eingeführt.

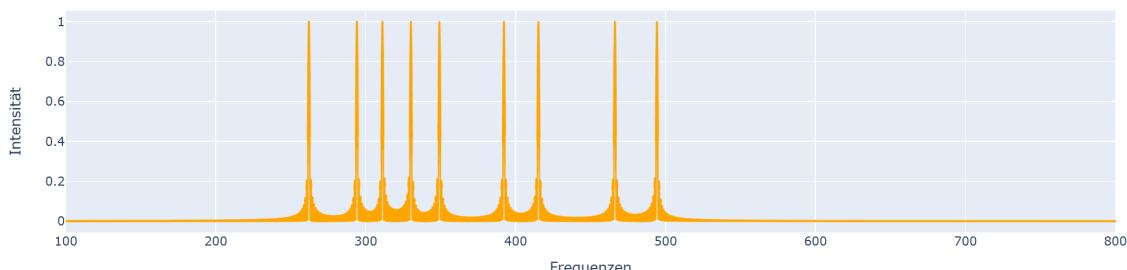
Fouriertransformation für Signale mit zeitlich variabler Frequenzzusammensetzung

Um den Übergang vom Audiosignal eines Dreiklanges zu einem Sprachsignal zu erleichtern, wird als Zwischenschritt eine Abfolge dreier verschiedener Dreiklänge betrachtet. Am Beispiel dieser Tonabfolge wird der Unterschied eines zeitlich konstanten Sprachsignals zu einem Audiosignal mit sich zeitlich ändernden Frequenzen thematisiert. Eine Dreiklangabfolge ist – ähnlich wie ein Sprachsignal – hörbar dynamischer, da die Mischung der Frequenzen nicht wie beim Dreiklang konstant ist, sondern sich mit der Zeit ändert. Aus mathematischer Sicht bedeutet das, dass sich die Frequenzraum-Darstellung eines Sprachsignals (im Gegensatz zu der eines Dreiklangs) im zeitlichen Verlauf ändert. Für die Erkennung eines Sprachsignals muss die Frequenzraum-Darstellung zu jedem Zeitpunkt bekannt sein.

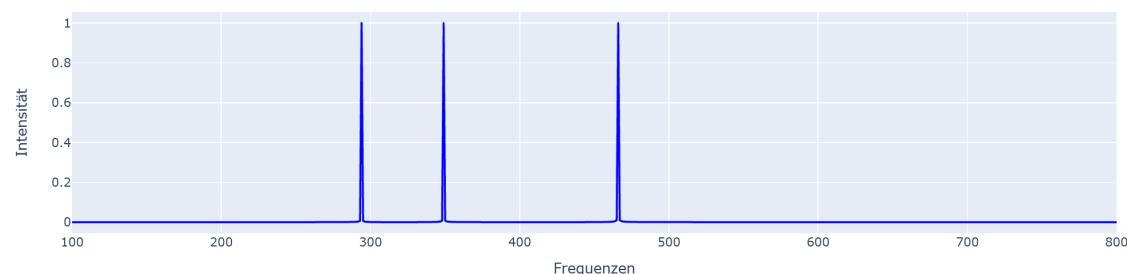
Die Dreiklangabfolge kann angehört und als Graph betrachtet werden (siehe Abbildung 19.2(a)). Die Konsequenz, die sich aus der variierenden Frequenzzusammensetzung für das Vorgehen bei der Fourier-Analyse ergibt, erarbeiten sich die Schülerinnen und Schüler eigenständig. Dafür werden mit der Fouriertransformation zunächst wie gewohnt die neun im Audiosignal enthaltenen Frequenzen ermittelt (drei Frequenzen pro Dreiklang) (siehe Abbildung 19.2(b)). Dass die einzelnen Frequenzen so allerdings nicht den Dreiklängen zuzuordnen sind, wird klar, wenn im nächsten Arbeitsauftrag die einzelnen Dreiklänge rekonstruiert werden sollen und anschließend angehört werden können. Anhand des bisherigen Wissens ist es nicht möglich, zu entscheiden, welche Frequenzen zu welchem Zeitpunkt in der Dreiklangabfolge abgespielt werden. Es ist daher nicht klar, zu welchem Dreiklang die einzelnen Töne gehören. Die Zusammensetzung der Dreiklänge kann daher nur erraten werden. Die rekonstruierte Dreiklangabfolge klingt damit höchstwahrscheinlich nicht so wie die ursprüngliche Dreiklangabfolge.



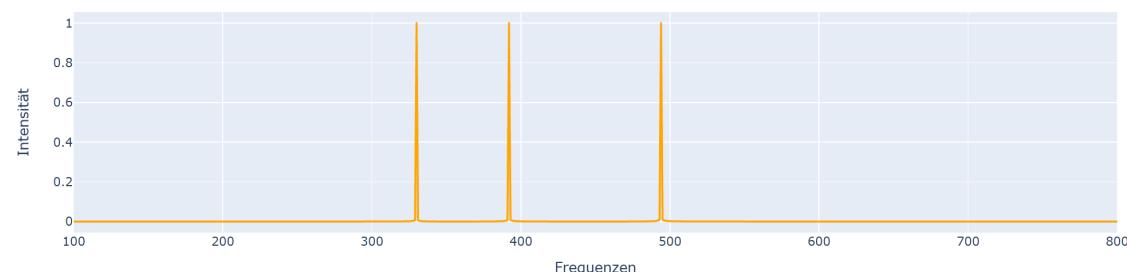
(a) Audiosignal der Dreiklangabfolge



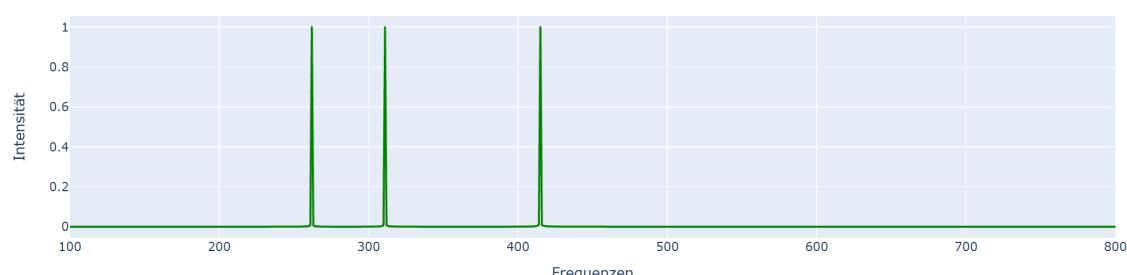
(b) Amplitudenspektrum der Dreiklangabfolge



(c) Amplitudenspektrum des ersten Dreiklangs



(d) Amplitudenspektrum des zweiten Dreiklangs



(e) Amplitudenspektrum des dritten Dreiklangs

Abbildung 19.2: Abfolge dreier Dreiklänge in verschiedenen Darstellungen

Die Lernenden haben daraufhin die Möglichkeit, die Fouriertransformation nur auf Teile des Signals anzuwenden. Sie haben die Aufgabe, das Signal in Zeitblöcke zu unterteilen, in denen sich die Zusammensetzung der einzelnen Sinustöne nicht ändert. Dafür legen sie die Anzahl der Zeitblöcke und die Dauer eines Blockes fest. Sie sind über die Länge der Audioaufnahme und den Graphen des Audiosignals zu ermitteln. Anschließend können die Dreiklänge über die Ausgabe der Amplitudenspektren der einzelnen Blöcke bestimmt werden (siehe Abbildungen 19.2(c), 19.2(d) und 19.2(e)). Auch hier kann über die Ausgabe der Audiosignale der originalen Dreiklangabfolge und der ausgewählten Dreiklangabfolge die Eingabe überprüft werden.

Im Lernmaterial wird der Begriff „Zeitblock“ deshalb verwendet, da dieser hervorhebt, dass das zeitliche Element eine bestimmte Dauer besitzt, wohingegen „Zeitschritt“ und „Zeitpunkt“ einen punktuellen Zeitwert ohne Dauer implizieren. Die Bedeutung der Zeitdauer ist insbesondere im Kontext der Fourieranalyse von Bedeutung, da hier deutlich wird, dass zur Ermittlung der enthaltenen Frequenzen die Funktion über einen Zeitraum hinweg analysiert werden muss und eine Momentaufnahme nicht ausreicht. Der Begriff „Zeitintervall“ betont ebenfalls die zeitliche Dauer, erweist sich jedoch als weniger geeignet, sobald nach der Fourieranalyse die diskretisierte Darstellung des Signals im Frequenzraum betrachtet wird. Der Begriff „Zeitblock“ vereint beide Konzepte und ermöglicht eine durchgehende konsistente Verwendung im gesamten Lernmaterial.

Die anhand einer einfachen Dreiklangabfolge erarbeitete Vorgehensweise wird anschließend bei einem Sprachsignal als komplexerem akustischen Signal, angewendet. Wegen der raschen Abfolge unterschiedlicher Tongemische in Sprachsignalen ist es bei deren Analyse erforderlich, zahlreiche kurze Blöcke zu verwenden. Bei einer Länge eines Audioausschnittes von 1024 Werten und einer Abtastrate von 22050 Hz ergeben sich bei einem zweisekündigen Audiosignal $\lceil \frac{22050}{1024} \cdot 2 \rceil = 44$ Blöcke. Das Spektrogramm kann von den Lernenden mithilfe eines Schiebereglers für verschiedene Zeitblöcke betrachtet werden. Hier wird bewusst auf die zwar übliche und übersichtliche, jedoch auch komplexe Darstellung des Spektrogramms in einem farbcodierten Zeit-Frequenz-Diagramm verzichtet, um niemanden zu überfordern. Stattdessen wird explizit die gleiche Darstellungsform gewählt wie bei der Dreiklangabfolge. Dabei werden die Amplituden der Frequenzen in Form eines Stabdiagramms für einen festen Zeitblock dargestellt (vgl. Abbildung 19.3). Mit dieser Art von Diagramm besteht bereits eine Vertrautheit durch den Unterricht. Verschiedene Zeitblöcke können durch Betätigen eines Schiebereglers ausgewählt werden.

Reduktion der Merkmale - Schritt für Schritt zur Mel-Filterung

Im Diagramm ist schnell ersichtlich, dass in einem realen Sprachsignal sehr viel mehr Frequenzen überlagert sind als bei einer simplen Dreiklangabfolge. Diese Frequenzen müssen bei der Erkennung des Sprachsignals alle weiter verarbeitet werden. Ein großes Problem dabei ist, dass die charakteristischen Teile des Sprachsignals von vielen anderen Frequenzen „zugedeckt“ sind. Dies sind Frequenzen, die zum Beispiel nicht zum Sprachsignal gehören, sondern Hintergrundgeräusche sind. In manchen Bereichen sind die Frequenzen auch unnötig genau aufgelöst. Dies macht die Analyse und Erkennung des Sprachsignals schwierig. Deshalb erhalten die Lernenden die Aufgabe, Ideen zu sammeln, wie die Frequenzen sinnvoll reduziert werden können. Im

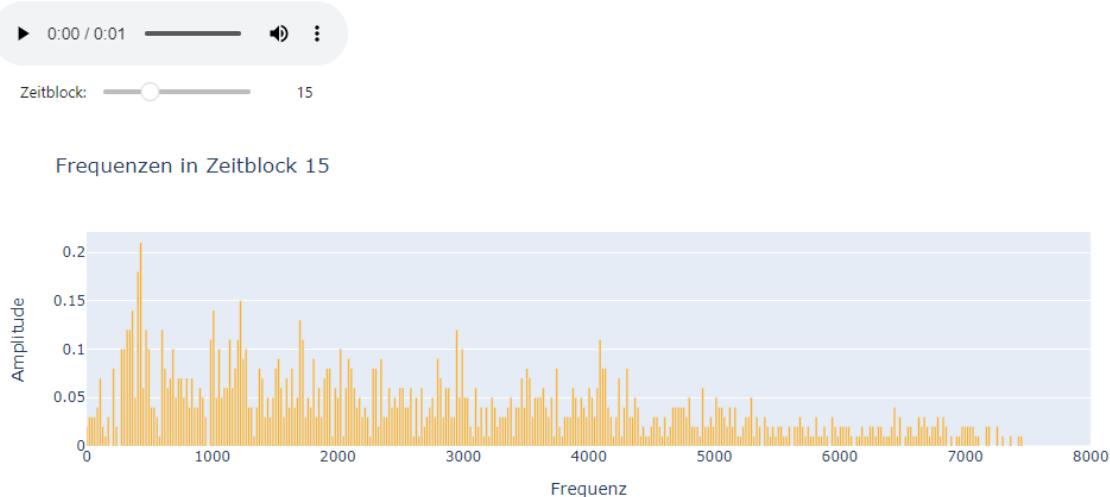


Abbildung 19.3: Bildschirmfoto des Amplitudenspektrums eines Audiosignals in verschiedenen Zeithöfen, auswählbar durch einen Schieberegler

Lernmaterial wird anschließend die Filterung mit einer Mel-Filterbank zur Reduktion der Frequenzen durchgeführt (vgl. Abschnitt 17.1.3). Sie wird mit den Lernenden nicht explizit durchgeführt. Stattdessen werden die grundlegenden Aspekte der Mel-Filterung den Lernenden in Form einzelner Maßnahmen, welche vom menschlichen Hörvorgang abgeleitet sind, näher gebracht. Folgende Maßnahmen werden dabei in den Blick genommen:

- (1) Es werden diejenigen Frequenzen eliminiert, die vom menschlichen Gehör nicht wahrgenommen werden können.
- (2) Mehrere benachbarte Frequenzen werden zusammengefasst, da Töne, deren Frequenzen sehr nahe beieinander liegen, nicht unterschieden werden können.
- (3) Die Frequenzen werden unterschiedlich stark gewichtet, da hohe Frequenzen vom menschlichen Gehör weniger differenziert wahrgenommen werden als niedrige Frequenzen.

Maßnahme (1)

Um Frequenzen zu entfernen, die vom menschlichen Gehör nicht wahrgenommen werden können, wird eine obere und untere Grenzfrequenz festgelegt. Diese orientieren sich an der psychologisch erfassten Sprachwahrnehmbarkeit und werden in der Spracherkennung typischerweise auf 100 Hz (untere Grenzfrequenz) bzw. 6800 Hz (obere Grenzfrequenz) festgelegt (M & Kopparapu, 2014, S. 4). Die Filterung der Frequenzen kann durch die Eingabe einer oberen und unteren Grenzfrequenz durchgeführt werden. Durch den sowohl visuellen Output in Form eines Stabdiagrammes als auch auditiven Output kann der Einfluss der Filterung bei unterschiedlichen Grenzfrequenzen auf das Sprachsignal untersucht werden (vgl. Abbildung 19.4). In der Aufgabe werden die Grenzfrequenzen der Sprachwahrnehmbarkeit werden zwar angegeben, die Lernenden werden aber explizit dazu aufgefordert auch andere Wer-

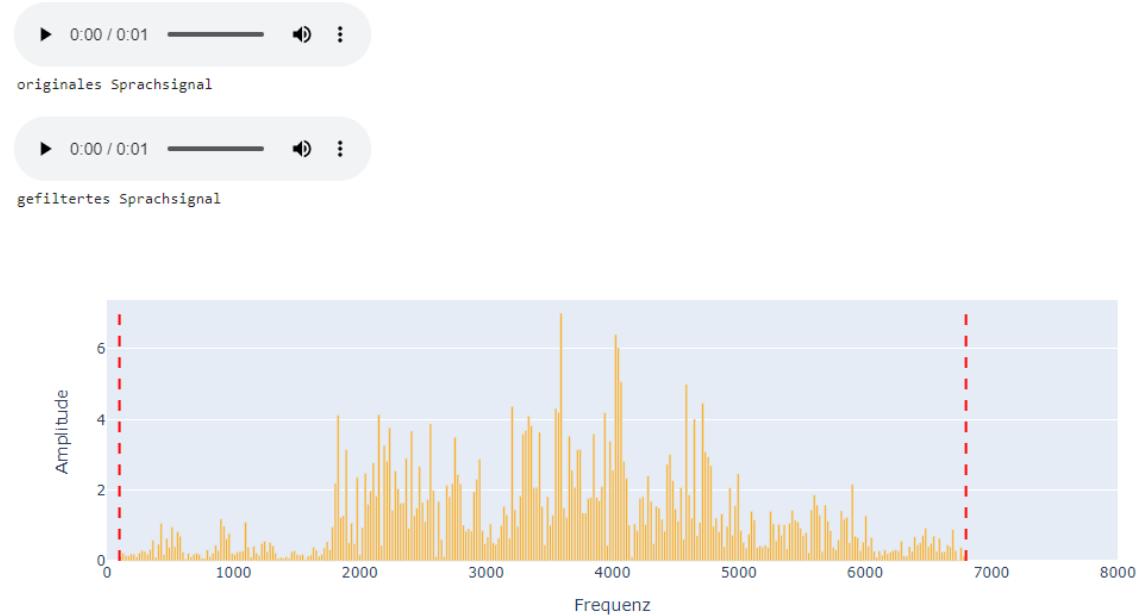


Abbildung 19.4: Bildschirmfoto des Amplitudenspektrums eines spezifischen Zeitblockes mit eingezeichneten Grenzfrequenzen und anhörbaren Audiodateien des originalen und gefilterten Audiosignals

te auszuprobieren. Bei einer unteren Grenzfrequenz von 1000 Hz und einer oberen Grenzfrequenz von 3000 Hz ist das Sprachsignal zum Beispiel nur noch schwer zu identifizieren. Werden jedoch die Grenzfrequenzen von 100 Hz bzw. 6800 Hz verwendet, so ist wie gewünscht kaum ein Unterschied zum ursprünglichen Audiosignal zu hören.

Maßnahme (2)

Für die weitere Reduktion der Daten, werden Frequenzen in kleinen Intervallen zusammengefasst. Hierbei werden den Lernenden verschiedene Möglichkeiten gegeben, wie die Amplitudenwerte eines Frequenzintervalls zusammengefasst werden können unter denen sie sich für eine begründet entscheiden sollen:

- Den Mittelwert der Amplituden der Frequenzen eines Frequenzintervall bilden.
- Die Summe der Amplituden der Frequenzen eines Frequenzintervall bilden.
- Die größte Amplitude aus dem Frequenzintervall auswählen.
- Die Amplitude der Frequenz, die genau in der Mitte des Frequenzintervall liegt, auswählen.

Nur eine der vier Möglichkeiten ergibt jedoch aus mathematischer und physikalischer Sicht Sinn. Wird die größte Amplitude aus dem Frequenzintervall ausgewählt, so werden die anderen Amplitudenwerte außer Acht gelassen. Das Frequenzintervall würde in diesem Fall lediglich die Ausprägung der Frequenz mit der größten Amplitude widerspiegeln.

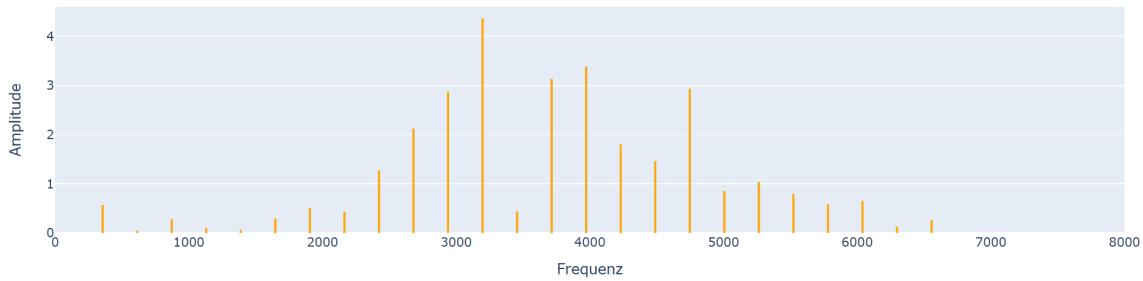


Abbildung 19.5: gemitteltes Amplitudenspektrum eines spezifischen Zeitblockes

Wird diejenige Amplitude ausgewählt, die genau in der Mitte des Frequenzintervall liegt, verhält es sich ähnlich. Da die Frequenzen im Intervall außerdem nicht nach der Größe ihres Amplitudenwertes geordnet sind, erscheint es willkürlich, einfach den mittleren Wert herauszugreifen. Die positive Eigenschaft der Robustheit gegen Ausreißer des mittleren Wertes einer geordneten Stichprobe (Median) entfällt damit ebenfalls. Der Ausreißer könnte in diesem Fall nämlich zufällig in der Mitte der Stichprobe liegen und würde das Ergebnis verfälschen. Aufgrund der Tatsache, dass diese Methode ebenfalls alle Amplitudenwerte, bis auf einen unberücksichtigt lässt, eignet sich auch diese Methode nicht, um die Frequenzen zusammenzufassen.

Bei der Addition aller Amplitudenwerte eines Frequenzintervalls werden alle Amplitudenwerte berücksichtigt. Allerdings entspricht die Summe dieser Werte nicht mehr der tatsächlichen Stärke, mit der dieses Frequenzintervall im Sprachsignal vorhanden ist. Erst mit der Division mit der Anzahl der Amplitudenwerte ergibt der Wert auch physikalisch Sinn. Er gibt an, mit welcher Stärke das Frequenzintervall im Sprachsignal vorhanden ist. Da auch bei der Mel-Filterung ein (gewichteter) Mittelwert gebildet wird, stimmt dieses Vorgehen mit einem in der Praxis gängigen Verfahren zur Filterung überein (vgl. Abschnitt 17.1.3). Die ermittelten Merkmale werden erneut in einem Stabdiagramm dargestellt (vgl. Abbildung 19.5). Auf die Darstellung als Säulendiagramm wird hier bewusst verzichtet. Die Gründe hierfür werden bei der nächsten Maßnahme deutlich.

Maßnahme (3)

Bisher wurden die Frequenzintervalle bei hohen und tiefen Frequenzen gleich breit gewählt. Da die Frequenzauflösung des menschlichen Gehörs mit höheren Frequenzen stark abnimmt, kann die Reduktion der Merkmale jedoch effektiver gestaltet werden. Die Lernenden sollen dazu die Breite der Frequenzintervalle aus verschiedenen Antwortmöglichkeiten so wählen, dass sie sich an der Frequenzauflösung des menschlichen Gehörs orientieren. Da hohe Frequenzen schlechter unterschieden werden können, sollten hier mehrere Frequenzen zusammengefasst werden. Die Frequenzintervalle sollten deshalb zu hohen Frequenzen hin breiter gewählt werden. Als Ausgabe erfolgt ein Diagramm der zusammengefassten Frequenzwerte. Auch hier wird an der Darstellung in Form eines Stabdiagrammes festgehalten (vgl. Abbildung 19.6). Es wird explizit kein Säulendiagramm verwendet, da den Säulen entsprechend ihrer Dicke automatisch eine unterschiedliche Wichtigkeit zugesprochen wird. Hingegen wird

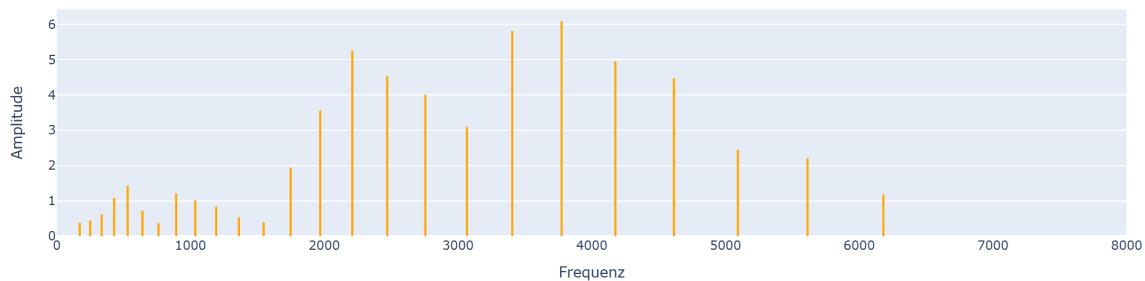


Abbildung 19.6: Amplitudenspektrum eines spezifischen Zeitblockes nach der Mel-Filterung

jeder Amplitudenwert in der weiteren Verarbeitung gleichwertig berücksichtigt und stellt ein Merkmal dar.

Dass die Breite der Frequenzintervalle nicht linear erhöht wird, sondern einer Mel-Skalierung folgt, wird im Lernmaterial lediglich benannt. Erarbeitet wird die Skalierung allerdings nicht explizit. Auch die Glättung durch eine Überlappung der Filter, sodass der Einfluss der Intervallgrenzen eliminiert wird, sowie die daraus resultierende dreieckige Form der Filter wird im Lernmaterial nicht thematisiert und lediglich im Hintergrund angewendet. So wird die mathematische Komplexität reduziert, und trotzdem kann die unterschiedliche Breite der Intervalle als mathematische Modellierung des psychoakustischen Effekts der abnehmenden Frequenzauflösung bei höheren Frequenzen thematisiert werden.

Da insgesamt viele Schritte zur Ermittlung der Merkmale durchgeführt werden, wird eine Rekapitulation dieser Schritte am Ende des dritten Arbeitsblattes durchgeführt und die Ordnung der einzelnen Schritte in die richtige Reihenfolge gefordert. Die Lernenden sollen sich überlegen, was sie in Arbeitsblatt 2 und 3 bisher in welcher Reihenfolge umgesetzt haben. Dies entspricht dem folgenden gesuchten Vorgang, bei welchem

1. eine Unterteilung der Sprachsignale in Zeitblöcke erfolgt,
2. Amplituden einzelner Frequenzen durch Fourier-Transformation bestimmt werden,
3. eine Entfernung besonders tiefer / hoher Töne stattfindet,
4. Frequenzen, die nahe beieinanderliegen, zu Frequenzintervallen zusammengefasst werden,
5. alle ermittelten Merkmale pro Zeitblock als ein Muster zusammengefasst werden.

19.2.4 Arbeitsblatt 4: Abstände zwischen Merkmalsvektoren

Das Ziel des vierten Arbeitsblatts (vgl. Anhang B.4) ist die Klassifizierung der in Arbeitsblatt 3 ermittelten Sprachmuster mithilfe des Nächster-Nachbar-Verfahrens

(vgl. Abschnitt 17.2.3). Dazu wird zunächst der euklidischen Abstand im mehrdimensionalen Raum erarbeitet. Anschließend wird ein Abstandsmaß für den Gesamtabstand zweier Muster zueinander ermittelt. Zuletzt wird das Nächster-Nachbar-Verfahren erläutert und das entwickelte Spracherkennungssystem erprobt.

Ⓜ Einordnung in den Modellierungskreislauf

Statt Dreiklängen soll ein tatsächliches Sprachsignal klassifiziert werden. Dazu werden die bisherigen Annahmen angepasst. Es werden Sprachaufnahmen von Einzelwörtern betrachtet. Dabei wird die Auswahl der Wörter auf wenige verschiedene Wörter begrenzt. Das vereinfachte Problem ist die sogenannte Einzelworterkennung, bei der im Vergleich zu fließend gesprochener Sprache jedes Wort einzeln als Sprachsignal vorliegt. Die Wortbedeutung eines Sprachsignals ist nicht mehr wie beim Dreiklang direkt aus dem Frequenzspektrum auslesbar. Stattdessen ist für die Erkennung eines neuen Sprachsignals ein Vergleich mit Sprachsignalen bekannter Wortbedeutung erforderlich. Auf mathematischer Ebene erfolgt dieser Vergleich anhand des euklidischen Abstand zwischen den zugehörigen Sprachmustern.

Zur Veranschaulichung der zu tätigen Klassifizierung wird im Plenum zunächst die geometrische Visualisierung der Sprachmuster als Punkte im Raum gezeigt. Zuerst wird ein Merkmal der Sprachmuster in einem ausgewählten Zeitblock als Punkt auf dem Zahlenstrahl dargestellt. Diese eindimensionale Darstellung erleichtert den Anschluss zu Arbeitsblatt 6. Hier ist es von Vorteil, nur ein Merkmal zu betrachten, um den zeitlichen Verlauf der Sprachmuster darstellen zu können. Anschließend werden zwei Merkmale der Sprachmuster in einem Zeitblock betrachtet und als Punkt in der Ebene visualisiert. Beim Übergang von der eindimensionalen zur zweidimensionalen Darstellung wird deutlich, dass jedes weitere Merkmal eine zusätzliche Dimension in der Darstellung erfordert.

Im Plenum werden verschiedene Maße für den Abstand der Sprachmuster gesammelt. Die Lernenden werden hier natürlich durch die gewählte geometrische Darstellungsform bei ihrer Ideenfindung beeinflusst. Aufgrund ihres Vorwissens zugängliche Abstandskonzepte sind die absolute Abweichung der Sprachmuster (vgl. (17.10)), der Winkel zwischen den zugehörigen Ortsvektoren und die daraus resultierende Kosinus-Ähnlichkeit sowie der euklidische Abstand der Sprachmuster (vgl. (17.9)). Im Plenum werden diejenigen Abstandsmaße besprochen, welche von den Lernenden genannt werden. Im Schulunterricht lernen die Schülerinnen und Schüler die euklidische Metrik als Abstand kennen. Es ist daher zu erwarten, dass der euklidische Abstand am häufigsten genannt wird. Außerdem im Schulunterricht auftreten können die Kosinus-Ähnlichkeit beispielsweise bei der geometrischen Herleitung des Skalarprodukts über den Kosinus (Schwarz & Herrmann, 2015, S. 212) oder die von der L1-Norm abgeleitete Manhattan-Metrik. In der Regel werden diese allerdings nicht explizit thematisiert. Mit dem vorhandenen Vorwissen ist es aber durchaus denkbar, dass die absolute Abweichung und der Winkel als Möglichkeiten für die Messung der Ähnlichkeit genannt werden. Mit den Schülerinnen und Schülern kann kurz darüber gesprochen werden, dass das ebenfalls denkbare Möglichkeiten sind, um

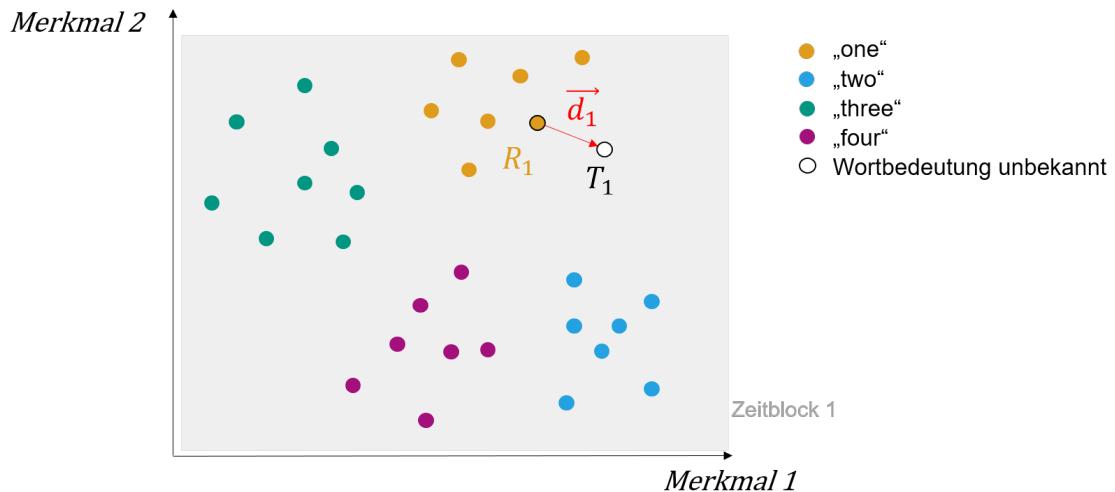


Abbildung 19.7: Abstand eines Referenzmusters zu einem Testmuster im Zweidimensionalen im ersten Zeitblock

die Ähnlichkeit zweier Vektoren zu ermitteln. Typische Ideen, die Lernende benannt haben, können in Kapitel 20 nachgelesen werden. Auf eine ausführliche Thematisierung unterschiedlicher Metriken wird aus Zeitgründen verzichtet und stattdessen mit der euklidischen Metrik weitergearbeitet, da so an das Vorwissen der Lernenden zur Berechnung des Abstands zwischen zwei Punkten in der Ebene oder im Raum angeknüpft werden kann (vgl. Abschnitt 19.3).

Der Abstand zweier Sprachmuster wird auf Arbeitsblatt 4 von den Lernenden bestimmt. Dabei wird mit dem zweidimensionalen Problem begonnen (vgl. Abbildung 19.7). Pro Zeitblock werden nur zwei Merkmale der Sprachmuster betrachtet, wobei der euklidische Abstand im ersten Zeitblock für zwei Sprachmuster bestimmt werden soll.

Anschließend werden drei Merkmale pro Sprachmuster und pro Zeitblock betrachtet. Diese lassen sich als Punkt im dreidimensionalen Raum auffassen. Erneut soll der Abstand bestimmt werden und die Formel zur Berechnung mit der Berechnung des Abstands zweier Punkte im Zweidimensionalen verglichen werden. Durch diese Analogiebetrachtung des Abstandsbegriffes in ebener und räumlicher Geometrie wird das Wissen über die ebene Geometrie gefestigt und gleichzeitig das Entdecken von Regelmäßigkeiten und Zusammenhängen geschult. Diese Regelmäßigkeiten können dann auch für die mehrdimensionale Geometrie verallgemeinert werden.

Zusammenhänge lassen sich dabei zum einen im geometrischen Vorgehen finden. Für die Abstandsbestimmung wird zunächst der Verbindungsvektor der beiden Punkte gebildet und anschließend dessen Länge ermittelt. Dafür wird der Verbindungsvektor in der zwei- und dreidimensionalen Geometrie in seine Orthogonal-Projektionen entlang der jeweiligen Koordinaten-Achsen zerlegt. Anschließend wird die Länge des Verbindungsvektors über das Anwenden des Satzes von Pythagoras für verschiedenen rechtwinklige Dreiecke, die sich durch die Projektionen ergeben, bestimmt. Unterstützung erhalten die Lernenden durch zu dieser Aufgabe zugehörige Tipps.

Diese unterstreichen die Zusammenhänge, indem sie in gleicher Reihenfolge sowohl in der ebenen als auch in der räumlichen Geometrie eine Schritt für Schritt konkreter werdende Anleitung für das geometrische Vorgehen bei der Abstandsbestimmung zur Verfügung stellen (vgl. Abbildung 19.8).

Zum anderen lassen sich Regelmäßigkeiten in der Formel für den euklidischen Abstand erkennen (vgl. Abbildung 19.8). Diese Regelmäßigkeiten in der algebraischen Vorgehensweise sind insbesondere für die Verallgemeinerung des Vorgehens auf die mehrdimensionale Geometrie hilfreich, in der eine anschauliche Bestimmung des Betrags des Verbindungsvektors nicht mehr möglich ist.

Bisher wurde lediglich ein einziger Zeitblock betrachtet und der Abstand zwischen dem neuen Sprachmuster und dem Referenzmuster für diesen Zeitblock berechnet. Tatsächlich sind die eingesprochenen Signale bei der Fourier-Transformation allerdings in 44 Zeitblöcke unterteilt worden. Die Abstände d_1, \dots, d_{44} in den weiteren Zeitblöcken können nach dem gleichen Prinzip wie in Zeitblock 1 bestimmt werden. Für den Vergleich der gesamten Sprachmuster ist es notwendig, einen Gesamtabstand zu bestimmen. Die Lernenden haben dazu erneut die Aufgabe, verschiedene Möglichkeiten für die Bestimmung eines Gesamtabstands gegeneinander abzuwägen. Folgende Auswahlmöglichkeiten stehen ihnen dabei zur Verfügung:

- $d_{ges} = d_1 - d_2 - \dots - d_{44}$
- $d_{ges} = d_1 + d_2 + \dots + d_{44}$
- $d_{ges} = d_1 - d_2 + d_3 - \dots - d_{43} + d_{44}$

Nur bei der Addition gehen die einzelnen Abstände in gleichem Maße in die Formel ein. Diese kann als das Zusammenfassen einzelner Werte gedeutet werden und ist daher die richtige Wahl für die Bestimmung eines Gesamtabstands.

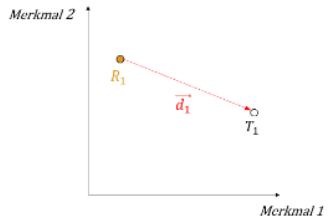
Um ein neues Sprachsignal zu erkennen, wird dessen Sprachmuster mit verschiedenen Referenzmustern verglichen, indem der Gesamtabstand des Sprachmusters zu jedem Referenzmuster ermittelt wird. Das Sprachsignal wird schließlich der Klasse zugeordnet, deren Sprachmuster den kleinsten Abstand zum neuen Sprachmuster haben. Bei diesem Minimierungsproblem handelt es sich um das in (17.12) formulierte Optimierungsproblem, welches im Lernmaterial zum besseren Verständnis für die Schülerinnen und Schüler statt mit einer mathematisch symbolischer Darstellung in sprachlicher Form erfolgt.

Das in Abschnitt 17.2.3 vorgestellte Nächster-Nachbar-Verfahren wird im Hintergrund ausgeführt. Die Lernenden erhalten als Ausgabe lediglich die über das Verfahren ermittelte Klassenzugehörigkeit sowie den Wert des kleinsten ermittelten Abstands zu einem Referenzmuster. Die Lernenden haben nun die Möglichkeit, die Methode mit vier echten Sprachsignalen anzuwenden und das Ergebnis zu reflektieren. Die Sprachsignale sollen angehört und ihre Wortbedeutung mit der ermittelten Klassenzugehörigkeit verglichen werden. Dabei fällt auf, dass nicht alle Sprachsignale korrekt klassifiziert werden. Eines der vier beispielhaften Sprachsignale, das im Vergleich weniger deutlich klingt, wird auch vom Spracherkennungssystem nicht korrekt erkannt. Mögliche Gründe können sein, dass zu wenige Referenzmuster zur

Tipps | Berechnung des Abstandes im Zweidimensionalen

Tipp 1

1. Bilde als erstes den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet.
2. Berechne anschließend den Betrag des Vektors \vec{d}_1 .



▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 2

1. Für den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet, gilt:

$$\vec{d}_1 = \begin{pmatrix} T_1[0] - R_1[0] \\ T_1[1] - R_1[1] \end{pmatrix}$$

2. Berechne den Betrag des Vektors \vec{d}_1 .

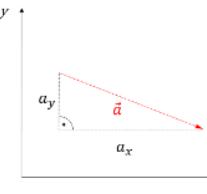
▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 3

Hier findest du Tipps um den Betrag eines Vektors zu berechnen.

Unter dem **Betrag eines Vektors \vec{a}** versteht man die Länge des zu \vec{a} gehörenden Pfeils.

1. Bestimme die Längen der gestrichelten Linien.
2. Berechne anschließend die Länge des roten Pfeils. Verwende dazu die Längen der gestrichelten Linien und den Satz des Pythagoras.

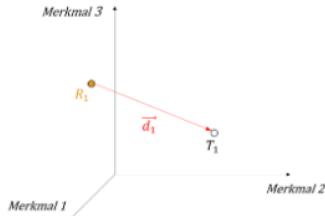


► Satz des Pythagoras (Zur Wiederholung des Satz des Pythagoras kannst du hier klicken.)

Tipps | Berechnung des Abstandes im Dreidimensionalen

Tipp 1

1. Bilde als erstes den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet.
2. Berechne anschließend den Betrag des Vektors \vec{d}_1 .



▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 2

1. Für den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet, gilt:

$$\vec{d}_1 = \begin{pmatrix} T_1[0] - R_1[0] \\ T_1[1] - R_1[1] \\ T_1[2] - R_1[2] \end{pmatrix}$$

2. Berechne den Betrag des Vektors \vec{d}_1 .

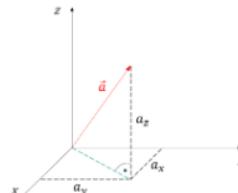
▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 3

Hier findest du Tipps um den Betrag eines Vektors zu berechnen.

Unter dem **Betrag eines Vektors \vec{a}** versteht man die Länge des zu \vec{a} gehörenden Pfeils.

1. Berechne die Länge der grün gestrichelten Linie. Nutze hierzu den Satz des Pythagoras.
2. Berechne anschließend die Länge des roten Pfeils. Verwende auch hier wieder den Satz des Pythagoras.



► Satz des Pythagoras (Zur Wiederholung des Satz des Pythagoras kannst du hier klicken.)

Tipps | Berechnung des Abstandes im Mehrdimensionalen

Tipp 1

1. Bilde als erstes den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet.
2. Berechne anschließend den Betrag des Vektors \vec{d}_1 .

▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 2

1. Für den Vektor \vec{d}_1 , der die beiden Punkte R_1 und T_1 verbindet, gilt:

$$\vec{d}_1 = \begin{pmatrix} T_1[0] - R_1[0] \\ T_1[1] - R_1[1] \\ \vdots \\ T_1[24] - R_1[24] \end{pmatrix}$$

2. Berechne den Betrag des Vektors \vec{d}_1 .

▼ Klicke hier, falls du einen weiteren Tipp benötigst

Tipp 3

Den Betrag eines Vektors erhält man, indem man seine Einträge quadriert, summiert und anschließend die Wurzel zieht.

Abbildung 19.8: Gegenüberstellung der Tipps zur Abstandsbestimmung im zwei-, drei- und mehrdimensionalen Raum

Verfügung stehen oder die Referenzmuster nicht repräsentativ genug sind, um auch verrauschte und/ oder Sprachsignale von ganz unterschiedlicher Stimmlage zu erkennen. Geometrisch anschaulich betrachtet bedeutet dies, dass eventuell keine oder zu wenige Punkte in der Nähe des neuen zu klassifizierenden Datenpunktes liegen. Dies spiegelt auch der Abstand wider, welcher bei der Fehlklassifikation zum nächsten Nachbarn ermittelt wurde. Im Vergleich zu den Abständen, die bei den drei anderen Klassifikationen zum nächsten Datenpunkt ermittelt wurden, ist dieser merklich größer.

Auch eine Überschneidung der Datenpunkt wolken kann zu Problemen bei der Klassifikation führen. Dies kann passieren, wenn sich die Referenzmuster verschiedener Klassen sehr ähnlich sind. Ein Nichtbeachten der zeitlichen Variabilität im Sprachverlauf könnte ebenfalls ein Grund für die falsche Klassifikation sein. Die Überlegungen der Lernenden zu möglichen Gründen für eine Fehlklassifikation können später im Plenum besprochen und durch den Input der Lehrkraft ergänzt werden.

Zuletzt soll das bisherige Vorgehen rekapituliert werden, indem die wichtigsten Schritte für die Entwicklung eines Spracherkennungssystems in die richtige Reihenfolge gebracht werden. Nacheinander werden dafür

1. das Sprachsignal eingelesen,
2. das Sprachsignal mit der Fouriertransformation analysiert,
3. aus dem Amplitudenspektrum Merkmale ermittelt und Referenzmuster gebildet,
4. das Muster des neuen Sprachsignals mit den Referenzmustern verglichen,
5. die Klassenzugehörigkeit des Referenzmusters mit kleinstem Abstand ausgewählt.

19.2.5 Arbeitsblatt 5: Bewertung des Spracherkennungssystems

Das Erproben des Spracherkennungssystems mit einzelnen Sprachsignalen motiviert die Definition geeigneter Qualitätsmaße. Hierzu lernen die Schülerinnen und Schüler auf Arbeitsblatt 5 (vgl. Anhang B.5) die Darstellung der Klassifikationsergebnisse in einer Wahrheitstabelle sowie die Qualitätsmaße Präzision und Genauigkeit kennen und ermitteln die zugehörigen Werte für das entwickelte Spracherkennungssystem. Anschließend werden die Klassifikationsergebnisse sowie die ermittelten Qualitätsmaße interpretiert und deren Aussagekraft untersucht.

Ⓜ Einordnung in den Modellierungskreislauf

In der auf das Arbeitsblatt 4 folgenden Plenumsdiskussion wird die Klassifizierung mit Mustervergleich als mathematisches Modell in den Modellierungskreislauf eingeordnet. Die vorhergesagten Klassenzugehörigkeiten, welche die zugehörigen Wortbedeutungen darstellen, ergeben sich als mathematische Lösung. Diese gilt es nun auf die Realsituation zu beziehen und die Ergebnisse systematisch zu validieren.

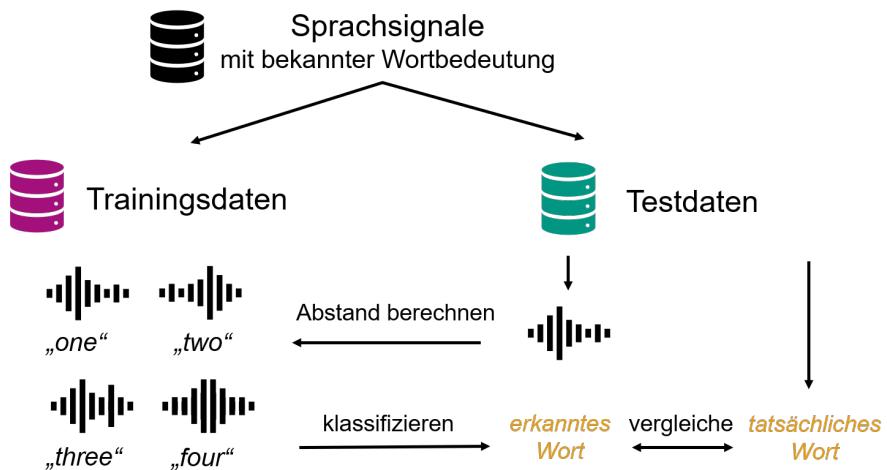


Abbildung 19.9: Rolle der Test- und Trainingsdaten bei der Klassifizierung von Sprachsignalen

Systematische Tests sind notwendig, um die Qualität eines Spracherkennungssystems zu beurteilen. Hierzu wird eine Strategie des überwachten Maschinellen Lernens vorgestellt, bei welcher Daten, deren Klassifikationsergebnis bereits bekannt sind, zunächst im Verhältnis eins zu vier in Test- und Trainingsdaten aufgesplittet werden. Abbildung 19.9 zeigt diese Aufteilung und veranschaulicht die Verwendung der Test- und Trainingsdaten im Kontext der Spracherkennung. Sie gibt den Lernenden somit eine Veranschaulichung einer datengetriebenen Modellentwicklung und -validierung und ordnet gleichzeitig die vorherigen Entwicklungsschritte des Spracherkennungssystems in den Gesamtzusammenhang ein. Im Lernmaterial werden 192 Sprachsignale als Trainingsdaten und 48 Sprachsignale als Testdaten genutzt, deren Muster nach dem vorher erarbeiteten Vorgehen automatisiert ermittelt werden. Anschließend wird der Abstand eines Sprachmusters aus den Testdaten zu jedem Referenzmuster gebildet, und anhand des kleinsten Abstands kann das Sprachmuster klassifiziert werden. Abschließend wird die erkannte Wortbedeutung mit der tatsächlichen Wortbedeutung verglichen und überprüft, ob sie übereinstimmen. Die Ergebnisse dieses Vergleichs werden in einer Wahrheitstabelle gespeichert, die den Lernenden zunächst in ihrer allgemeinen Form vorgestellt und anschließend für die Ergebnisse der mit dem entwickelten Spracherkennungssystem klassifizierten 48 Sprachsignale ausgegeben wird. Die Klassifikation der Testdaten und die Erstellung der Wahrheitstabelle geschieht im Hintergrund.

Dabei wird insofern eine kontextualisierte Wahrheitstabelle verwendet, als anstelle der Begriffe „tatsächliches Klassenlabel“ und „klassifiziertes Klassenlabel“, die Bezeichnungen „erkanntes Wort“ als Überschrift der Spalten und die Bezeichnung „tatsächliches Wort“ als Überschrift der Zeilen genutzt werden. Die Verwendung einer kontextualisierten Wahrheitstabelle führt im Vergleich zu einer Standard Wahrheitstabelle nämlich laut Shen et al. (2020, S. 13) zu einem wesentlich besseren Verständnis. Zudem zeigten Shen et al. (2020, S. 13), dass Lernenden das Lesen

		Wahrheitstabelle			
		one	two	zero	three
tatsächliches Wort	one	6	2	0	3
	two	2	6	3	2
	three	0	2	5	1
	four	1	1	1	13
		one	two	three	four
		erkanntes Wort			

Abbildung 19.10: Wahrheitsmatrix für die Klassifikation mit einfachem Mustervergleich

und Interpretieren der Wahrheitsmatrix mit absoluten Werten leichter fällt als mit Anteilswerten. Im Lernmaterial werden in der Wahrheitsmatrix daher nicht die prozentualen Anteile der klassifizierten Sprachsignale eingetragen, sondern die absolute Anzahl der klassifizierten Sprachsignale (vgl. Abbildung 19.10).

Zudem wird auf eine sorgfältige Auswahl an Begriffen aus der KI-Literatur geachtet. Ziel ist es, nur die essentiell notwendigen Fachbegriffe zu verwenden, um einen kognitiven Überforderung zu verhindern. Auf einen Begriff wie „Label“, der leicht durch den deutschen, intuitiv verständlicheren Begriff „Klassenzugehörigkeit“ ersetzt werden kann, wird daher verzichtet. Weitere Begriffe wie „Trainings- und Testdaten“, „Genauigkeit“, „Präzision“ sowie der Begriff „Wahrheitstabelle“, die als relevant für die Beschreibung der Vorgänge angesehen werden, werden aus dem Englischen übersetzt und als Fachbegriffe vorgestellt. Anstelle des wörtlich übersetzten Begriffs „Wahrheitsmatrix“ wird jedoch der Begriff „Wahrheitstabelle“ verwendet, da das Wort „Matrix“ nicht bekannt sein dürfte.²

Die Wahrheitstabelle soll anschließend hinsichtlich der Fragen

- Wie viele Signale wurden insgesamt richtig klassifiziert?
- Wie viele Signale wurden insgesamt falsch klassifiziert?
- Welches Wort wurde am häufigsten richtig erkannt?
- Welches Wort wurde am seltensten richtig erkannt?
- Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

²Matrizen sind laut Bildungsplan kein Inhalt des Mathematikunterrichts an Gymnasien in Baden-Württemberg (vgl. Ministerium BW, 2016d).

untersucht werden. Von den 48 Sprachsignalen wurden 30 korrekt und 18 nicht korrekt klassifiziert. Das Wort „four“ wurde am häufigsten und das Wort „three“ am seltensten richtig erkannt. Die meisten Verwechslungen traten zwischen den Wörtern „two“ und „three“ auf. In zwei Fällen wird das Wort „three“ fälschlicherweise als „two“ erkannt und in drei Fällen wird das Wort „two“ fälschlicherweise der Klasse „three“ zugeordnet. Betrachtet man die Aussprache der beiden Wörter, so fällt auf, dass sich diese im Vergleich zu den anderen Wörter phonetisch ähnlich sind. Beide Wörter beginnen mit einem stimmlosen Konsonanten, gefolgt von einem langen stimmhaften Vokal. Diese Ähnlichkeit könnte tatsächlich die hohe Übereinstimmung bei der Vorhersage ihrer Klassenzugehörigkeit erklären. Um die Ursachen jedoch genauer zu ergründen, wären eine Untersuchung weiterer Faktoren sowie Tests des Spracherkennungssystems mit einer größeren Datenbasis erforderlich.

Durch die Bearbeitung dieser Aufgabe setzen sich die Lernenden intensiv mit der Wahrheitstabelle auseinander und es wird überprüft, ob sie diese Wahrheitstabelle lesen können. Zugleich leiten die erste und zweite Frage direkt zu den bereits in Abschnitt 17.4 vorgestellten Qualitätsmaßen über. Die Qualitätsmaße Genauigkeit und Präzision werden im Lernmaterial zunächst allgemein als Formel eingeführt. Anschließend sollen sie für die vorliegende Wahrheitstabelle bestimmt werden. Auch hiernach erfolgt die Aufgabe zur Interpretation der berechneten Werte für die Qualitätsmaße. Dazu sollen die Fragen

- Was bedeutet es, wenn der Wert der Präzision nahe null beziehungsweise nahe eins liegt? Bei welchen Wörtern ist dies der Fall?
- Sind die Ergebnisse der Klassifikation zufriedenstellend? Falls nein, wo müssten die Ergebnisse noch verbessert werden?

beantwortet werden. Liegt die Präzision einer Klasse nahe null bzw. eins, so bedeutet dies, dass nahezu keine bzw. nahezu alle als diese Klasse klassifizierten Sprachsignale richtig klassifiziert sind. Keine der für die Klassen berechnete Präzision ist nahe null. Die Präzision für Klassen „one“ und „four“ liegen mit 0,67 und am 0,68 nächsten bei eins, sind damit aber auch noch nicht nahe genug bei eins, sodass hier zwar schon viele aber noch nicht fast alle Wörter die als „one“ beziehungsweise „four“ erkannt werden, auch korrekt klassifiziert wurden. Bei einer Präzision der Klassen „two“ und „three“ von 0,55 und 0,56 ist die Klassenzugehörigkeit ungefähr jedes zweiten Mal, wenn ein Wort als „two“ oder „three“ klassifiziert wird, nicht korrekt erkannt worden. Auch ein Blick auf die Genauigkeit von 0,62 zeigt, dass die Ergebnisse insgesamt noch nicht zufriedenstellend sind und Verbesserungen zwar insbesondere für die Unterscheidung der Klassen „two“ und „three“, aber auch im Allgemeinen für die Unterscheidung aller Klassen notwendig sind.

Für eine reflektierte Bewertung der Aussagekraft der Qualitätsmaße und ein vertieftes Auseinandersetzen mit der Wahrheitstabelle soll anschließend ein weiteres fiktives Spracherkennungssystem bewertet werden. Dieses System wird auf einem zweiten Testdatensatz getestet. Entscheidend ist, dass dieser Datensatz von 48 Sprachsignalen 34 Mal das Wort „one“ enthält. Auf Basis der gegebenen Klassifizierungsergebnisse erstellen die Lernenden eine Wahrheitstabelle, mit deren Hilfe Rückschlüsse auf das Spracherkennungssystem gezogen werden sollen. Dieses System klassifiziert alle

Sprachsignale ausnahmslos als Klasse „one“. Dieses Vorgehen ist objektiv betrachtet sinnfrei, dennoch rückt die Ermittlung der Genauigkeit das Spracherkennungssystem in ein anderes Licht. Die Genauigkeit liegt bei

$$\frac{34}{48} \approx 0,71.$$

Die Lernenden sollen daher auch die Präzision ermitteln und anschließend eine begründete Auswahl zwischen dem von ihnen entwickelten Spracherkennungssystem und demjenigen, das kategorisch alles als „one“ klassifiziert, treffen. Da die Präzision der Klassen „two“, „three“ und „four“ bei null liegt, sollte trotz der hohen Genauigkeit die Entscheidung zugunsten des im Workshop entwickelten Systems ausfallen. Die Lernenden sollen auf Basis dieser Entscheidung die Aussagekraft der Qualitätsmaße diskutieren. Insbesondere bei einer Unausgewogenheit der einzelnen Klassen in den Testdaten sagt die Genauigkeit allein wenig über die Qualität eines Spracherkennungssystems aus. Daher ist es sinnvoll, stets mehrere Qualitätsmaße zu berücksichtigen. Zusätzlich sollen die Lernenden erkennen, dass für einen Vergleich zweier Systeme, diese im besten Fall auf denselben Testdaten getestet werden sollten. Der Einfluss des Testdatensatzes auf den Wert der ausgewählten Qualitätsmaße, welcher von Hazzan und Mike (2023, S. 2014) als typische Hürde mit wenig statistischem Hintergrundwissen dargestellt wird, kann so direkt am Beispiel erfahrbar gemacht werden.

Abschließend sammeln die Lernenden Vorschläge zur Verbesserung des entwickelten Spracherkennungssystems, die im Plenum diskutiert werden. Zu möglichen Verbesserungen zählen:

- Erweiterung des Trainingsdatensatzes: Durch die Vergrößerung des Trainingsdatensatzes stehen mehr Referenzmuster zur Verfügung, was bei den vier Datenpunktewolken zu einer höheren Punktdichte führt. Dadurch würden außerdem eventuell noch mehr unterschiedliche Aussprechvarianten der vier Zahlenwörter erfasst werden, deren Referenzmuster die Punktwolken vergrößern würde. Der Nachteil hiervon ist, dass der Zeitaufwand der Klassifizierung beträchtlich steigt, da der Abstand zu jedem Referenzmuster bestimmt werden muss.
- Berücksichtigung mehrerer Referenzmuster (k -nächste-Nachbarn-Verfahren): Statt nur einem Referenzmuster könnten die k ähnlichsten Referenzmuster bei der Klassifizierung berücksichtigt werden. Ob dies tatsächlich eine Verbesserung bei dem hier verwendeten Datensatz darstellt, müsste jedoch experimentell überprüft werden.
- Verbesserung der Datenqualität: Die Qualität datenbasierter Verfahren hängt maßgeblich von den zugrunde liegenden Daten ab. Eine bessere Audioqualität würde zu genaueren Klassifizierungsergebnissen führen. Dies könnte durch die Reduzierung von Hintergrundgeräuschen und die Verwendung hochwertigerer Mikrofone bei der Aufnahme der Trainings- und Testdaten erreicht werden.
- Berücksichtigung zeitlicher Unterschiede in Sprachsignalen: Aufgrund der variablen Geschwindigkeit, mit der Laute ausgesprochen werden, und möglichen

zeitlichen Verzögerungen bei den Aufnahmen, ist es sinnvoll, bei der Klassifizierung der Sprachsignale zeitliche Unterschiede zu berücksichtigen. Diese etablierte Methode in der Spracherkennung wird im sechsten Arbeitsblatt behandelt.

Im Unterrichtsgespräch werden nach der Sammlung von Verbesserungsvorschlägen der Lernenden die oben genannten Verbesserungsmöglichkeiten exemplarisch vorgestellt. Die Auswahl berücksichtigt die Vorkenntnisse der Schülerinnen und Schüler und konzentriert sich auf gut nachvollziehbare Ansätze. Letztlich sollte in diesem Gespräch individuell auf die gemachten Vorschläge eingegangen und die Vorstellung der Verbesserungsmöglichkeiten an diese angepasst werden.

19.2.6 Arbeitsblatt 6: Zeitliche Anpassung der Sprachmuster

Auf Arbeitsblatt 6 (vgl. Anhang B.6) wird die zeitliche Komponente der Merkmale mit berücksichtigt und das Muster als Zeitreihe – das heißt als zeitlich geordnete Folge der Merkmalswerte – aufgefasst. Dazu werden zunächst der zeitliche Versatz der einzelnen Zustände zweier konkreter Sprachsignale zueinander betrachtet und verschiedene Darstellungsformate dieses Versatzes untersucht. Anschließend wird an einem konkreten Beispiel das Verfahren des Dynamic Time Warping erarbeitet. Ein Vergleich des ursprünglichen Spracherkennungssystems und desjenigen mit Dynamic Time Warping kann anschließend mithilfe der erarbeiteten Qualitätsmaße erfolgen.

④ Einordnung in den Modellierungskreislauf

Aus didaktischen Gründen wird die Reihenfolge der Evaluation und der zeitlichen Anpassung der Sprachmuster im Vergleich zur formalen Analyse der mathematischen Hintergründe in Kapitel 19 geändert. Um sich zunächst auf die wesentlichen Schritte zu fokussieren, wird im Vergleich zu dem in Abbildung 17.1 dargestellten Ablauf der Schritt der dynamischen Zeitanpassung zunächst übersprungen. So kann der Modellierungszyklus eines ersten einfachen Spracherkennungssystems zuerst in seiner Gänze durchlaufen werden und eine Evaluation dessen erfolgen, bevor weitere Modifikationen dieses Systems folgen. Die Evaluation ist zudem ein wichtiger Schritt, um überhaupt festzustellen, dass Verbesserungen notwendig sind. Die zeitliche Anpassung kann infolgedessen als Modellverbesserung eingeführt werden. Dieses Vorgehen erlaubt den Vergleich der Systeme mit und ohne dynamische Zeitanpassung anhand der Qualitätsmaße, sodass die Nützlichkeit der zeitlichen Anpassung herausgestellt werden kann. Zugleich wird gewährleistet, dass die Lernenden nicht durch eine zu große Menge an neuen Informationen überfordert werden und den Überblick über die Schritte, die für die Erstellung eines Spracherkennungssystems notwendig sind, verlieren.

Um das Verfahren des Dynamic Time Warping mit den Lernenden zu erarbeiten, wird die Vereinfachung vorgenommen, dass pro Zeitblock statt 25 Merkmalen nur ein Merkmal betrachtet wird. Dies erleichtert die Repräsentation eines Musters als



Abbildung 19.11: Das erste Merkmal der Muster *A* und *B* aufgetragen über den Zeitblöcken

Zeitreihe und die Veranschaulichung des Optimierungsverfahrens, ohne jedoch die Gültigkeit des Verfahrens für mehrdimensionale Merkmalsvektoren einzuschränken. Zunächst sollen die Probleme des bisherigen Verfahrens an den beiden Beispilmustern aus Abbildung 19.11 thematisiert werden. Dabei fällt auf, dass die beiden Sprachsignale optisch sehr ähnliche Muster aufweisen. Muster *B* scheint ab Zeitblock 4 lediglich um einen Zeitblock gegen Muster *A* verschoben zu sein. Wird allerdings der Gesamtabstand über Gleichung (17.11) bestimmt unter der Annahme, dass nur das erste Merkmal vorliegt, so ergibt sich der Gesamtabstand zu

$$D(A, B) = 0 + 0 + 0 + 1 + 2 + 1 + 1 + 0 + 0 = 5.$$

Da der Abstand als Maß für die Ähnlichkeit dient, würde der berechnete Abstand den Sprachmustern eine geringe Ähnlichkeit zuschreiben, was im Kontrast zur optischen Ähnlichkeit der Sprachmuster steht. Anhand dieses Beispiels sollen die Lernenden erkennen, dass für die Verwendung der euklidischen Metrik als Maß für die Ähnlichkeit zunächst eine zeitliche Anpassung der Sprachsignale erforderlich ist. Dazu werden zeitlich zueinander passende Bereiche in den Mustern übereinander gelegt und anschließend der Gesamtabstand bestimmt. Hierfür wird das Muster als Zeitreihe betrachtet. Der Begriff Zeitreihe wird jedoch nicht verwendet; stattdessen wird davon gesprochen, die Sprachmuster nicht mehr nur Zeitblock für Zeitblock zu betrachten, sondern den zeitlichen Verlauf in den Blick zu nehmen.

Die dynamische Zeitanpassung wird anschließend für die Beispilmuster in Abbildung 19.11 veranschaulicht. Eine zeitliche Anpassung, bei welcher der Gesamtabstand minimiert wird, ist für dieses Beispiel noch mit bloßem Auge ersichtlich. Dazu wird zunächst mithilfe der Darstellung in Abbildung 19.11 erläutert, in welchem Zeitblock eine zeitliche Streckung des Musters *A* bzw. des Musters *B* stattfinden muss, um die Sprachmuster aufeinander anzupassen. Die Erläuterung wird durch eine animierte Grafik veranschaulicht. Zudem wird den Lernenden die Darstellung des Zuordnungspfades als Liste aus Tupeln der Zustände aus Muster *A* und Muster *B* sowie die Darstellung des Zuordnungspfades im Graphen (vgl. Abbildung 19.12) vorgestellt.

Anschließend soll eine Zuordnung für die zeitliche Anpassung zweier neuer Muster bestimmt werden, die ebenfalls durch zwei lokale zeitliche Streckungen und damit durch bloßes Hinsehen gefunden werden kann. Die Lernenden werden explizit aufge-

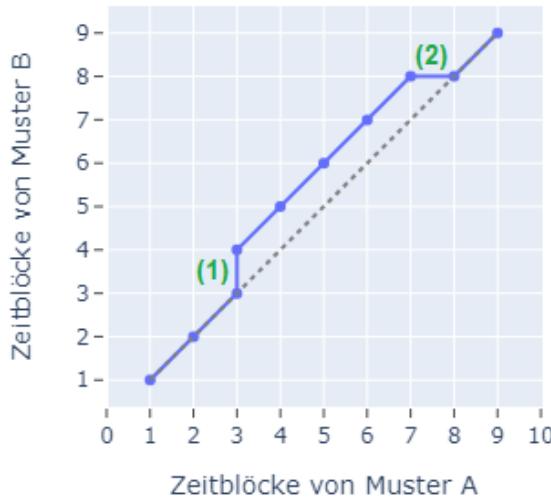


Abbildung 19.12: Optimaler Zuordnungspfad der Anpassung der Muster A und B

fordert, unterschiedliche Zuordnungen zu testen, um zu erkennen, dass die Lösung des vorliegenden Optimierungsproblems nicht eindeutig ist.

Wie in Abschnitt 17.3.1 bereits erläutert wird in der Literatur häufig der Begriff der Dynamic Time Warping Distanz verwendet, obwohl es sich bei diesem Längenmaß nicht um einen metrischen Abstand handelt. Im Lernmaterial wird daher der Begriff „Abstand“ beziehungsweise „Distanz“ im Zusammenhang mit dem Dynamic Time Warping explizit nur für die bereits angepassten Sprachmuster verwendet. Die zeitliche Anpassung und das Bilden des Gesamtabstands zwischen den Zeitblöcken der zeitlich aufeinander angepassten Sprachmuster werden als zwei voneinander getrennte Schritte betrachtet. Dies wird beispielsweise im erklärenden Text vor Aufgabe 2 d explizit benannt: „So ergibt sich der Zuordnungspfad, und zugleich wird direkt der Gesamtabstand zwischen den zeitlich aufeinander angepassten Sprachmustern bestimmt.“ (vgl. Anhang B.6).

Das Festlegen von Randbedingungen wird anschließend dadurch motiviert, dass die Zuordnung gewisse Eigenschaften der aufgenommenen Sprachsignale berücksichtigen soll. Die Mathematisierung dieser Bedingungen soll von den Lernenden vorgenommen werden. Dazu werden die folgenden Eigenschaften der aufgenommenen Sprachsignale genannt:

- Die zeitliche Reihenfolge der Laute einer Äußerung soll erhalten bleiben, sodass nicht plötzlich eine vom ursprünglichen Signal verschiedene Aneinanderreihung der Laute stattfinden kann und ein komplett neues Wort entsteht. Eine Änderung der zeitlichen Reihenfolge darf somit nicht möglich sein.
- Keine größeren Teile des Musters und damit z.B. ein ganzer Laut dürfen übersprungen werden.
- Die beiden Sprachsignale starten und enden zum gleichen Zeitpunkt.

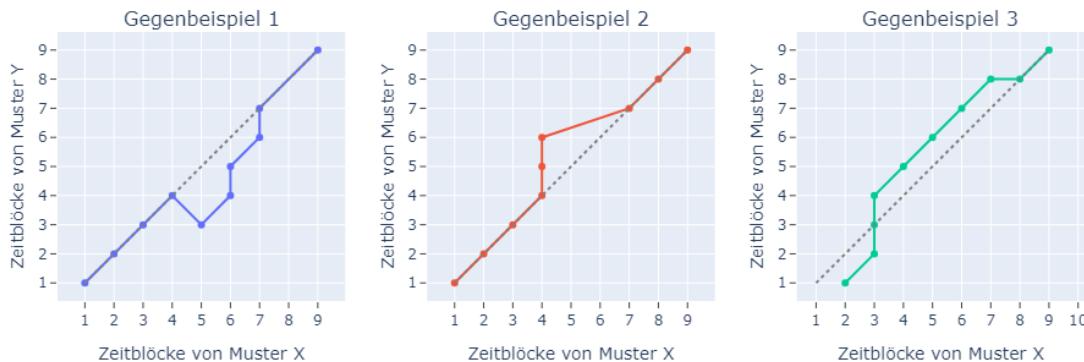


Abbildung 19.13: Beispiele von Zuordnungspfaden, bei denen in (1) die Kontinuitätsbedingung, in (2) die Monotoniebedingung und in (3) die Randwertebedingung verletzt werden

Diese sollen, wie bereits in Abschnitt 17.3.1 geschehen, den mathematischen Formulierungen der Randbedingungen Kontinuität (vgl. (17.14)), Randwerte (vgl. (17.15)) und Monotonie (vgl. (17.16)) zugeordnet werden. Zudem soll die Zuordnung dreier Gegenbeispiele in Form dreier Graphen von Zuordnungspfaden zu den Randbedingungen einen anschaulichen Zugang geben und zugleich das Verständnis fördern (vgl. Abbildung 19.13).

Als differenzierendes Element dient eine Zusatzaufgabe. In dieser sollen die Lernenden zunächst für eine Musterlänge von drei Zeitblöcken und anschließend für eine Länge von neun Zeitblöcken die Anzahl der möglichen Zuordnungen ermitteln. Dies dient als Vorbereitung, um die Notwendigkeit eines systematischen Verfahrens zur Bestimmung einer Zuordnung mit minimalem Gesamtabstand herauszuarbeiten.

Für drei Zeitblöcke kann die Anzahl der Zuordnungen noch durch strategisches Abzählen erfolgen. Es existiert genau ein Zuordnungspfad, der zwei Diagonalen enthält, sechs Zuordnungspfade mit einer Diagonale (davon zwei Zuordnungspfade, bei denen die Diagonale im ersten Schritt platziert ist, zwei Zuordnungspfade, bei denen der erste Schritt nach rechts erfolgt und zwei zwei Zuordnungspfade bei welchen der erste Schritt nach oben erfolgt) und sechs Zuordnungspfade ohne Diagonale. Daraus ergeben sich insgesamt 23 verschiedene Zuordnungen.

Bei einer Musterlänge von neun Zeitblöcken wird das Abzählen der Zuordnungspfade zunehmend komplex. Daher ist es erforderlich, eine allgemeingültige Formel auf Basis kombinatorischer Überlegungen aufzustellen. Eine sinnvolle Vorgehensweise besteht darin, sich an der zuvor genutzten Methode zu orientieren, indem die Zuordnungspfade nach der Anzahl der enthaltenen Diagonalen gruppiert werden. Um von Zustand $(1, 1)$ zum letzten Zustand (N, N) zu gelangen, werden ohne diagonale Schritte $N - 1$ Schritte nach oben und $N - 1$ Schritte nach rechts benötigt. Für jeden diagonalen Schritt, der sowohl einen Schritt nach oben als auch einen nach rechts ersetzt, verringert sich die Anzahl der verbleibenden separaten Schritte nach oben und rechts jeweils um eins. Bei d diagonalen Schritten verbleiben demnach $N - 1 - d$ Schritte nach rechts und genauso viele nach oben. Die Gesamtanzahl an Schritten

einschließlich diagonaler Bewegungen ist

$$N - 1 - d + N - 1 - d + d = 2(N - 1) - d.$$

Für eine feste Anzahl an enthaltenen Diagonalen ergibt sich die Anzahl der Pfade, indem die diagonalen Schritte und anschließend die Schritte nach rechts verteilt werden. Die Schritte nach oben werden in den verbleibenden Schritten getätig. Um die d diagonalen Schritte auf die insgesamt $2(N - 1) - d$ Schritte zu verteilen gib es $\binom{2 \cdot (N - 1) - d}{d}$ Möglichkeiten. Dies entspricht dem Ziehen der d diagonalen Schritten aus den Gesamtschritten ohne Zurücklegen und ohne Unterscheidung der diagonalen Schritte (Kombination ohne Wiederholung). Es verbleiben noch $2 \cdot (N - 1) - d - d = 2 \cdot (N - 1 - d)$ Schritte. Für die Wahl der Schritte nach rechts aus den verbleibenden Schritten gibt es $\binom{2 \cdot (N - 1 - d)}{N - 1 - d}$ Möglichkeiten. Ein Summieren über die Anzahl an diagonalen Schritten, die im Zuordnungspfad enthalten sind, liefert die Anzahl

$$P(N) = \sum_{d=0}^{N-1} \binom{2 \cdot (N - 1) - d}{d} \cdot \binom{2 \cdot (N - 1 - d)}{N - 1 - d}$$

der möglichen Zuordnungspfade. Bei einer Musterlänge von 9 Zeitblöcken ergeben sich somit $P(9) = 265729$ mögliche Zuordnungen. Diese hohe Anzahl verdeutlicht die Notwendigkeit eines systematischen Ansatzes, um die Zuordnung mit dem minimalen Gesamtabstand zu finden, da das bloße Ausprobieren aller Zuordnungen ineffizient wäre.

Das Verfahren des Dynamic Time Warpings wird anschließend am Beispiel zweier Muster wie in Abschnitt 17.3.2 vorgestellt. Für die ersten beiden Schritte wird das Vorgehen detailliert erläutert. Alle weiteren Schritte sollen von den Lernenden durchgeführt werden. Als ikonische Darstellungsform wird die in Abschnitt 17.3.2 vorgestellte Tabelle 17.5(b) verwendet. Diese wird anschließend von den Lernenden mit den weiteren Abstandswerten befüllt. Bei Bedarf werden sie dabei durch Tipps mit weiteren Beispielrechnungen zur Bestimmung zweier weiterer Abstandswerte in der Tabelle unterstützt.

Abschließend wird das Spracherkennungssystem mit Dynamic Time Warping getestet, und die Testergebnisse werden erneut in Form einer Wahrheitstabelle ausgeben. Die Lernenden sollen anschließend die auf Arbeitsblatt 5 kennengelernten Qualitätsmaße ermitteln sowie eine begründete Entscheidung für eines der beiden Spracherkennungssysteme (mit bzw. ohne DTW) treffen. Da alle Qualitätsmaße für das Spracherkennungssystem mit Modellverbesserung besser ausfallen (vgl. Tabelle 18.1), sollte die Wahl auf das Spracherkennungssystem mit Dynamic Time Warping fallen.

Diese Modellverbesserung soll beispielhaft eine typische Strategie für das Lösen komplexer Probleme in der Mathematik, die dynamischen Programmierung aufzeigen. Zudem wird an diesem Beispiel deutlich, dass Mathematik eine moderne, sich wandelnde Wissenschaft ist, deren Entwicklungen und Erkenntnisse auch in den letzten Jahrzehnten noch bedeutend voranschreiten.

Zum Abschluss erhalten die Lernenden die Gelegenheit, beide Spracherkennungssysteme selbst zu erproben. Dazu nehmen sie ein Sprachmemo auf, in dem sie eines der vier Zahlenwörter einsprechen. Dieses Wort wird anschließend von beiden Systemen klassifiziert. Zudem wird untersucht, inwiefern die Aufnahmequalität die Erkennungsleistung beeinflusst. Die daraus gewonnenen Erkenntnisse fließen in die abschließende Diskussion ein, in der die Grenzen der Spracherkennungssysteme und die zugrunde liegenden Ursachen thematisiert werden. So stellen etwa laute Hintergrundgeräusche oder eine zu große Entfernung vom Mikrofon klare Einschränkungen dar, da sie das Spektrogramm und somit das Sprachmuster so stark verändern, dass eine korrekte Zuordnung zu einer Klasse nicht mehr zuverlässig möglich ist. Auch das eingeschränkte Vokabular ist eine wesentliche Begrenzung der Anwendbarkeit. Ein solches System, wie es im Workshop entwickelt wird, findet daher vor allem in Anwendungen mit wenigen Befehlen Verwendung, etwa bei automatischen Telefonabfragen oder Pick-to-Voice-Systemen in Lagerhäusern (Euler, 2006, S. 65).

Bevor abschließend im Plenum ein zusammenfassender Überblick über die notwendigen Schritte zur Entwicklung eines Spracherkennungssystems gegeben wird, folgt eine kritische Reflexion dieser Systeme aus ethischer Perspektive. Gemeinsam mit den Schülerinnen und Schülern werden potenziell problematische beziehungsweise ethisch fragwürdige Einsatzmöglichkeiten gesammelt. Dazu gehört beispielsweise der Missbrauch der Spracherkennung als Abhörwerkzeug oder der Einsatz in Chatbots und humanoiden Robotern, wo sie unseren Alltag erleichtern können, aber auch fälschlicherweise als Ersatz für echte soziale Interaktionen wahrgenommen werden könnten. Darüber hinaus werden Aspekte wie Datenschutz und nicht repräsentative Datensätze als ethisch bedenkliche Aspekte bei der Entwicklung von Spracherkennungssystemen diskutiert.

19.3 Curriculare Anbindung

Dieser Abschnitt soll zeigen, dass im Lernmaterial zahlreiche Inhalte aufgegriffen werden, die in den Lehrplänen verankert sind. Es wird die curriculare Anbindung an den Bildungsplan Baden-Württembergs im Fach Mathematik (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024) sowie übergeordnet an die Bildungsstandards für den mittleren Schulabschluss im Fach Mathematik (Kultusministerkonferenz, 2022) beschrieben. Es wird außerdem gezeigt, dass sich Bezüge zu weiteren Schulfächern, insbesondere im Schulfach Musik, Physik, NwT und Informatik finden. Auch hier wird der Bildungsplan von Baden-Württemberg als Grundlage verwendet, da die Workshops in diesem Bundesland durchgeführt wurden.

Anbindung an Curricula des Fachs Mathematik

Das Lernmaterial zur Spracherkennung ist problembezogen konzipiert, sodass die dort behandelten mathematischen Inhalte gezielt auf die Lösung des Problems ausgerichtet sind. Dabei werden mathematische Kompetenzen verwendet, die die Lernenden in verschiedenen Jahrgangsstufen erwerben.

Dazu zählt der Umgang mit trigonometrischen Funktionen und die Deutung der Wirkung einzelner Parameter trigonometrischer Funktionen auf deren Funktions-

graphen. Diese Inhalte werden in Jahrgangsstufe 9/ 10 unter der Leitidee *funktionaler Zusammenhang* erlernt (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 38) und im Lernmaterial im ersten und zweiten Arbeitsblatt bei der Modellierung von Tönen und der Behandlung der Fouriertransformation am Beispiel von Dreiklängen aufgegriffen.

Indem die Transformation mit der Mel-Filterbank als eine Mittelwertbildung unterschiedlich breiter Frequenzbänder eingeführt wird, kann außerdem an das Vorwissen aus Jahrgangsstufe 5 und 6 der Leitidee *Daten und Zufall* angeknüpft werden. In dieser Jahrgangsstufe erlangen Schülerinnen und Schüler die Fähigkeit, das arithmetische Mittel einer Häufigkeitsverteilung zu bestimmen und im Sachkontext zu interpretieren (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 25; Kultusministerkonferenz, 2022, S. 21 - 22). Sie lernen außerdem den Umgang mit verschiedenen Darstellungsformen von Daten kennen, wie beispielsweise Stabdiagramme, die im Lernmaterial zur Darstellung der Spektrogramme genutzt werden.

Des weiteren wenden die Lernenden Inhalte der ebenen Geometrie bei der Klassifizierung der Sprachmuster an. Dabei spielen insbesondere die geometrische Interpretation von Vektoren als Punkte im Raum, die Berechnung des Abstands zwischen zwei Punkten in der Ebene oder im Raum sowie die Anwendung des Satzes von Pythagoras eine zentrale Rolle. Diese Themen sind typische Bestandteile des Curriculums in den Jahrgangsstufen 9 und 10 und unter den Leitideen *Raum und Form* sowie *Messen* verankert (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 36 - 37; Kultusministerkonferenz, 2022, S. 17, 20).

Zudem bietet das Lernmaterial die Möglichkeit, die Nutzung und Interpretation von Mehrfeldertafeln zu schulen. Sachverhalte mit Vierfeldertafeln zu untersuchen ist ebenfalls Teil des Curriculums und wird in Jahrgangsstufe 9 / 10 unter der Leitidee *Daten und Zufall* aufgeführt (Ministerium für Kultur, Jugend und Sport Baden-Württemberg, 2024, S. 37; Kultusministerkonferenz, 2022, S. 22). Mehrfeldertafeln als Verallgemeinerung der Vierfeldertafeln treten zwar weder in den Bildungsstandards noch im Bildungsplan des Gymnasiums Baden-Württembergs auf, deren Behandlung wird allerdings von Mathematikdidaktikern und -didaktikerinnen gefordert (Krüger, Sill & Sikora, 2015, S. 194). Dabei wird explizit benannt, dass das Lesen und Interpretieren von Mehrfeldertafeln anhand komplexer Beispiele geübt werden soll.

Die Erarbeitung und Anwendung des Nächster-Nachbar-Verfahrens sowie des Dynamic Time Warping Verfahrens bieten zudem die Möglichkeit, algorithmisches Denken zu fördern. Algorithmen sind unter der Leitidee *Zahl und Operation* in den Bildungsstandards verankert. Diese Leitidee umfasst das „[W]ählen, [B]eschreiben und [B]ewerten [von] Vorgehensweisen und Verfahren, denen Algorithmen bzw. Kalküle zu Grunde liegen und [das] [Aus]führen diese[r]“ (Kultusministerkonferenz, 2022, S. 16).

Insgesamt baut das Lernmaterial zur Spracherkennung auf den Inhalten des Mathematikunterrichts der Jahrgangsstufen 5 bis 10 auf. Es eignet sich daher ab Jahrgangsstufe 10, nachdem die Inhalte zur Vektorrechnung behandelt wurden. Das Lernma-

terial bietet die Möglichkeit, im Sinne des Spiralprinzips von Bruner und Hartung (1973) mathematische Inhalte, die in den genannten früheren Jahrgangsstufen behandelt wurden, zu vertiefen oder in neuen Kontexten erneut zu erschließen.

Anbindung an Curricula des Fachs Physik

Die im Lernmaterial behandelte Modellierung von Tönen ist ein typisches Thema im Physikunterricht der Jahrgangsstufe 7/ 8. Im Bildungsplan wird dies unter der Kategorie *Optik und Akustik* mit

„Die Schülerinnen und Schüler können optische und akustische Phänomene experimentell untersuchen. Sie trennen zunehmend zwischen ihrer Wahrnehmung und deren physikalischer Beschreibung. [...] Zur Beschreibung der Ausbreitung von Licht beziehungsweise Schall verwenden sie geeignete Modelle.“ (Ministerium BW, 2022, S. 16)

gefasst. Das Lernmaterial greift diese Inhalte auf den ersten beiden Arbeitsblättern auf. Es schult Schülerinnen und Schüler, „akustische Phänomene beschreiben [zu können] (Lautstärke, Tonhöhe, Amplitude, Frequenz)“ (Ministerium BW, 2022, S. 16), was ebenfalls explizit als Kompetenz im Bildungsplan verankert ist.

In der Oberstufe werden Schwingungen häufig in größerem Umfang behandelt, wobei der Fokus insbesondere auf akustischen Phänomenen wie dem Doppler-Effekt liegt (Ministerium BW, 2022, S. 36). Dafür ist ein fundiertes Verständnis der Grundlagen zur Beschreibung von Tönen unerlässlich. Ein sinnvoller Einstieg in diese Thematik kann daher eine Wiederholung der relevanten Konzepte sein, was über den Workshop erfolgen kann.

Anbindung an Curricula des Fachs Musik

Im Musikunterricht wird in der Unter- und Mittelstufe die Tonlehre behandelt. Dies umfasst die Bestimmung von Intervallen und darauf folgend die Bestimmung von Dreiklängen sowohl in Dur als auch in Moll in Jahrgangsstufe 5/ 6 (Ministerium BW, 2016b, S. 16). Der Workshop kann hier als Wiederholung dienen.

Zudem sollen Schülerinnen und Schüler in Jahrgangsstufe 7/ 8 die Prinzipien der Kangerzeugung von Instrumenten und der menschlichen Stimme näher gebracht und dabei insbesondere die Funktionsweise des menschlichen Atem- und Stimmaparates erforscht werden (Ministerium BW, 2016b, S. 19). Die Sichtweise mit der im Lernmaterial die von der menschlichen Stimme erzeugten Geräusche betrachtet wird, kann hierzu als sinnvolle Ergänzung dienen.

Anbindung an Curricula des Fachs Informatik

Auch der Lehrplan des Fachs Informatik bietet vielfältige Anknüpfungspunkte. Mit der geplanten Bildungsreform befindet sich das Fach Informatik im Wandel. Das Auslaufen des Fachs IMP (Informatik, Mathematik, Physik) wird von der Einführung eines eigenständigen Schulfachs „Informatik und Medienbildung“ begleitet (Ministerium BW, 2024b), dessen Inhalte sich vermutlich an den bisherigen Informatikanteilen des Fachs IMP orientieren. Entsprechend werden hier sowohl Bezüge zum Informatik-Anteil des Fachs IMP als auch zu den Lehrplaninhalten des Informatikunterrichts in der Oberstufe aufgezeigt.

Der Informatikunterricht soll Schülerinnen und Schüler dazu befähigen

„charakteristische Merkmale verschiedener Kommunikationsformen (Mensch-Mensch, Mensch-Maschine, Maschine-Maschine) auf Gemeinsamkeiten und Unterschiede analysieren und deren gesellschaftliche Auswirkungen bewerten“ (Ministerium BW, 2020, S. 13; Ministerium BW, 2016a, S. 17).

Das vorliegende Lernmaterial greift diese Zielsetzung auf, indem es die komfortabelste Form der Mensch-Maschine-Interaktion für den Menschen thematisiert: die Spracherkennung. Die Lernenden analysieren, wie sich maschinelle Spracherkennung vom intuitiven Sprachverständnis beim Menschen unterscheidet, und diskutieren am Ende des Workshops auch potenzielle Risiken dieser Technologie. Damit wird ein direkter Bezug zu den im Informatiklehrplan festgelegten Kompetenzen hergestellt. Auch die Datensicherheit, die ebenfalls im Informatik thematisiert werden soll (Ministerium BW, 2020, S. 23; Ministerium BW, 2016a, S. 33), ist ein Thema, das mit dem Lernen in der abschließenden kritischen Reflektion aufgegriffen werden kann. Zudem wird das algorithmische Denken insbesondere mit der Durchführung des Dynamic Time Warping Verfahrens gefördert, wobei wichtige Lösungsstrategien wie das Zerlegen in Teilprobleme aufgezeigt werden.

Das Thema Künstliche Intelligenz tritt derzeit explizit weder im Mathematik noch im Informatik Lehrplan auf. Dem soll allerdings mit der neuen Bildungsreform entgegengewirkt werden. Wie bereits oben erwähnt, wird ab dem Schuljahr 2025/26 gleichzeitig mit der Einführung von G9 auch das Pflichtfach „Informatik und Medienbildung“ durchgängig von Klasse 5 bis Klasse 11 eingeführt werden (Ministerium BW, 2024b). In diesem Fach sollen Kompetenzen im Bereich Informatik, Künstliche Intelligenz und Medienbildung verankert werden (Ministerium BW, 2023b). Das hier entwickelte Lernmaterial würde sich aufgrund der inhaltlichen Überschneidung gut für den Einsatz in diesem Fach eignen.

Anbindung an Curricula des Fachs NwT

Im Fach „Naturwissenschaft und Technik (NwT)“ werden Themen und Problemstellungen, denen die Lernenden in ihrer Alltagswelt begegnen, aus den Blickwinkeln aller Naturwissenschaften sowie der Technik fächervernetzt betrachtet. Die Sprachverarbeitung stellt ein solches Problem dar. Sie wird im Lernmaterial zwar vorrangig aus Sicht der Mathematik betrachtet, dennoch ergeben sich einige Anknüpfungspunkte an den NwT Lehrplan. Dies betrifft insbesondere die Verarbeitung von Informationen mithilfe von Sensoren, was im Lernmaterial die Sprachaufnahme darstellt. Im Bildungsplan wird dies wie folgt aufgegriffen:

„Der Mensch kann mithilfe seiner Sinnesorgane Signale aus der Umwelt schnell aufnehmen. Technische Sensoren übernehmen die gleichen Aufgaben und ermöglichen eine objektive Signalerfassung. [...] Durch den Vergleich der Funktionsweise von Sinnen und Sensoren erkennen die Schülerinnen und Schüler Parallelen und Unterschiede bei der Signal- und Informationsaufnahme in Natur und Technik.“ (Ministerium BW, 2016c, S. 25 - 26)

In Jahrgangsstufe 8/ 9 soll außerdem die Auswertung von Daten behandelt und dabei insbesondere die Darstellung von Frequenzen in einem Spektrum thematisiert werden (Ministerium BW, 2016c, S. 28).

20 Erfahrungswerte und didaktische Reflexion

20.1 Bisherige Erfahrungen

Um eine erste Beurteilung zur Eignung des Lehr- und Lernmaterials für mathematische Modellierungsprojekte zu erhalten, wurde dieses in verschiedenen Modellierungsveranstaltungen im Rahmen des Projekts CAMMP durchgeführt. Auf Grundlage der dabei gesammelten Erfahrungen wurden die Materialien in mehreren Iterationsschritten weiterentwickelt. Ziel dieser Erprobungen war es, eine vorläufige Einschätzung hinsichtlich des Interesses und der Motivation der Lernenden, der Angemessenheit des Schwierigkeitsgrades sowie des erzielten Fortschritts vorzunehmen. Dazu wurden die Lernenden bei der Durchführung beobachtet, und es fanden Umfragen zur Selbsteinschätzung nach der Durchführung der Modellierungsveranstaltung statt. Es sei jedoch hervorgehoben, dass für fundierte Aussagen über den Lernzuwachs und den langfristigen Lernerfolg weiterführende Untersuchungen, wie Leistungstests, eine Auswertung der Antworten zu den einzelnen Aufgaben des Lernmaterials oder Ähnliches erforderlich sind.

In Tabelle 20.1 sind alle durchgeführten Modellierungsveranstaltungen und deren Teilnehmendenzahlen einzeln aufgeführt. Bisher wurde das Lernmaterial insgesamt von 38 Schülerinnen und Schülern der gymnasialen Jahrgangsstufe 11 bearbeitet. Die drei teilnehmenden Kurse kamen von allgemeinbildenden Gymnasien aus der Karlsruher Umgebung. Die Modellierungstage fanden an unterschiedlichen Orten statt, und zwar einer an der jeweiligen Schule und die beiden anderen an der Universität.

Beobachtungen und daraus resultierende Anpassungen des Lernmaterials

Es werden nun zentrale Beobachtungen dargelegt, die während der Durchführungen gemacht wurden. Dabei wird auf die Antworten der Schülerinnen und Schüler während der Plenumsphase eingegangen, und es werden die beobachteten Lernschwierigkeiten beschrieben. Gleichzeitig wird dargelegt, welche Änderungen am Material vorgenommen wurden, um den beobachteten Schwierigkeiten zu begegnen.

Die Erfahrung mit den ersten beiden Arbeitsblättern soll hier kurz beschrieben werden, da die Entwicklung dieser Arbeitsblätter nicht Teil dieses Dissertationsprojekts war. Stattdessen wurden sie lediglich an den veränderten Kontext angepasst, und es

Tabelle 20.1: Tabellarische Übersicht über durchgeführte Modellierungsprojekte mit dem Lernmaterial zur Spracherkennung

Veranstaltung	Anzahl Lernende	Jahrgangsstufe
Modellierungstag mit Leistungskurs Mathematik eines allgemeinbildenden Gymnasiums	19	11
Modellierungstag mit Basiskurs Mathematik eines allgemeinbildenden Gymnasiums	13	11
Modellierungstag mit Leistungskurs Mathematik eines allgemeinbildenden Gymnasiums	6	11

fanden Überarbeitungen unter anderem im Design statt. In allen Durchführungen konnten die Schülerinnen und Schüler die trigonometrischen Funktionen als Funktionentyp zur Beschreibung von Tönen in dem zu einer angeschlagenen Stimmgabel gehörenden Amplituden-Zeit-Diagramm erkennen. Die Bearbeitung der ersten beiden Arbeitsblätter bereitete kaum Schwierigkeiten, vermutlich weil die behandelten mathematischen Inhalte zu trigonometrischen und linearen Funktionen sowie die Aufgabenstellungen sehr schulnah sind. Dass die abhängige Variable im Lernmaterial nicht wie aus dem Schulunterricht gewohnt mit x sondern mit t bezeichnet wurde, da sie die Zeit repräsentiert, sorgte teilweise für Verwirrung. Dies konnte jedoch schnell im Gespräch geklärt werden.

Bezüglich des dritten Arbeitsblatts konnte beobachtet werden, dass die Notwendigkeit der Fensterung erfasst und im Plenum korrekt erklärt werden konnte. Auch bei der Reduzierung der Frequenzen durch das Festlegen von Grenzfrequenzen und das Bilden von Mittelwerten der einzelnen Frequenzintervalle sind keine Lernhürden aufgefallen. Bei der ersten Durchführung fiel jedoch beim Rundgang durch die Gruppen auf, dass bei den Ankreuzaufgaben 4a und 4b (siehe Abbildung 20.1) manchmal ohne vorheriges Nachdenken Antworten ausgewählt wurden. Um ein unreflektiertes Ausprobieren der Antwortmöglichkeiten zu unterbinden, wurde anschließend auf diesem Arbeitsblatt und auch den nachfolgenden Arbeitsblättern ein Überprüfbrowser bei Ankreuzaufgaben eingefügt. Erst nach Betätigen des Überprüfbrowsers wird die Antwort überprüft und eine Rückmeldung ausgegeben.

In der anschließenden Plenumsphase fiel schnell auf, dass das Erkennen von Sprache komplexer ist als das Identifizieren eines Tons oder Dreiklangs. Schwierigkeiten wie unterschiedliche Stimmlagen, Dialekte, Betonungen, Sprechgeschwindigkeiten und Hintergrundgeräusche wurden von den Lernenden angesprochen.

Im Anschluss wurde gemeinsam erarbeitet, wie Sprachsignale dennoch erkannt werden können. Es wurde erklärt, dass die Merkmale eines Zeitblockes als Punkte im Raum aufgefasst werden können und die Erkennung von Sprachsignalen durch den Vergleich mit Sprachsignalen, deren Wortbedeutung bekannt ist, erfolgt. Ähnlichen

b) Breite der Frequenzintervalle anpassen

Entscheide, wie die Breite der Frequenzintervalle gewählt werden soll.

► Tipp

...

Mit höheren Frequenzen nimmt die Breite der Frequenzintervalle ab.

Mit höheren Frequenzen nimmt die Breite der Frequenzintervalle zu.

Bei niedrigen Frequenzen sind die Frequenzintervalle schmal, werden dann breiter und zu hohen Frequenzen hin wieder schmal.

Überprüfen

Abbildung 20.1: Bildschirmfoto der Aufgabe 4b als Beispiel für den Einsatz eines Überprüfbuttons

Sprachsignalen wird dann die gleiche Wortbedeutung zugeordnet. Es folgte eine Diskussion darüber, wie diese Ähnlichkeit gemessen werden kann. In allen Durchführungen erkannten die Teilnehmenden, dass die Amplituden der Frequenzen Zeitblock für Zeitblock verglichen werden müssen, um sie als Punkte im Raum zu interpretieren und Abstände zwischen ihnen berechnen zu können. In zwei der drei Durchführungen wurde außerdem die Idee benannt, den euklidischen Abstand der Merkmalsvektoren in den einzelnen Zeitblöcken zweier Sprachsignale als Indikator für die Ähnlichkeit zweier Sprachsignale zu verwenden. Auf die Frage, wann die Sprachmuster als ähnlich zu bezeichnen sind, antwortete eine Schülerin mit „dann, wenn der Abstand am geringsten ist in den einzelnen Zeiten“¹. Als zusätzliche Möglichkeit wurde in einer Durchführung von einem Lernenden die absolute Abweichung genannt.

Die Erarbeitung des euklidischen Abstands für die mehrdimensionale Geometrie mithilfe des vierten Arbeitsblatts bereitete den meisten keine Schwierigkeiten. Nach der ersten Durchführung fiel jedoch auf, dass einige den Überblick über die erlernten Methoden und deren Rolle im Spracherkennungssystem verloren hatten. Daher wurde am Ende des vierten Arbeitsblatts eine Aufgabe hinzugefügt, in der die einzelnen Schritte zur Entwicklung eines Spracherkennungssystems in die richtige Reihenfolge gebracht werden sollten.

Anschließend sollte das entwickelte Spracherkennungssystem bewertet werden. Bei der Sammlung von Bewertungsideen wurden diverse Ansätze genannt:

- Anzahl der richtig/falsch klassifizierten Datenpunkte: Es wird gezählt, wie häufig das eingesprochene Sprachsignal richtig erkannt wurde. In der Plenumsdiskussion argumentierten die Lernenden, dass dieser Wert für den Vergleich zweier Spracherkennungssysteme nicht sinnvoll ist, weil man nicht wisse, ob beide Male gleich viele Sprachsignale getestet werden.
- Genauigkeit: Es wird der Anteil der richtig erkannten Sprachsignale aus den insgesamt eingesprochenen Sprachsignalen bestimmt. Dieser Ansatz wurde als

¹Rechtschreib- und Interpunktionsfehler wurden in diesem und den nachfolgenden Schülerzitaten zur besseren Lesbarkeit korrigiert.

sinnvoller erachtet, weil er die Gesamtanzahl der getesteten Sprachsignale mit einbezieht.

- Fehlerrate: Es wird der Anteil der falsch erkannten Sprachsignale aus den insgesamt eingesprochenen Sprachsignalen bestimmt. Dieser Ansatz wurde in nur einer der Durchführungen genannt.

Die Präzision wurde nicht als Qualitätsmaß genannt. Beim Rundgang durch die Teams schienen die Lernenden allerdings keine Probleme mit der Definition oder der Berechnung der Präzision zu haben.

Sowohl beim Rundgang durch die Kleingruppen als auch bei der anschließenden Plenumsdiskussion zeigte sich, dass die Mehrheit in der Lage war, gefragte Informationen, wie beispielsweise die Anzahl der richtig klassifizierten Signale, aus der im Lernmaterial dargestellten Wahrheitstabelle abzulesen. In der Besprechung des Arbeitsblatts wurde zudem ersichtlich, dass die Schülerinnen und Schüler die Problematik eines unausgewogenen Testdatensatzes erkannten und es als sinnvoll erachteten, nicht nur ein Qualitätsmaß zu betrachten. Ein Schüler beschrieb die Problematik der Genauigkeit bei unausgewogenen Testdaten wie folgt:

„Wenn der Testdatensatz sehr unbalanciert ist, ist es nicht so gut, nur die Genauigkeit anzuschauen. Das haben wir gesehen, weil in einem Testdatensatz ‚one‘ 34 mal vorkommt und die anderen Wörter viel weniger. Das Spracherkennungssystem das alles als ‚one‘ erkennt, hatte eine hohe Genauigkeit, obwohl es ja eigentlich schlecht war. Also hier sollte man den Testdatensatz mehr ausgleichen oder nicht nur die Genauigkeit berechnen.“

Im Anschluss wurden im Plenum Ideen zur Verbesserung des Spracherkennungssystems gesammelt. Dabei brachten die Lernenden die folgenden sinnvollen Vorschläge ein:

- Die Verwendung einer größeren Menge an Trainingsdaten, um ein breiteres Spektrum an Sprachsignalen zum Vergleich bereitzustellen.
- Anstelle nur des ähnlichsten Referenzmusters könnten die k ähnlichsten Referenzmuster berücksichtigt werden, entsprechend dem Ansatz des k -nächsten-Nachbarn-Verfahrens.
- Es könnte eine Berechnung aller Abstände zwischen dem neuen Datenpunkt und den Referenzmustern vorgenommen werden, gefolgt von der Ermittlung des Mittelwertes der Abstände für jede Klasse. Auf diese Weise ließe sich die Klasse mit dem insgesamt geringsten mittleren Abstand zum neuen Datenpunkt bestimmen.

Arbeitsblatt 5 stellte alle vor größere Herausforderungen. Insbesondere Aufgabe 2b war zunächst so konzipiert, dass die Eigenschaften gesprochener Sprache vorgegeben wurden und daraus eigenständig Randbedingungen für die Zuordnungsfunktion der Sprachmuster abgeleitet werden sollten. Aufgrund des hohen Transferaufwands fiel es den meisten jedoch schwer, die Aufgabe zu lösen, sodass nahezu alle Unterstützung benötigten. Daher wurde die Aufgabe im Unterricht gemeinsam an der Tafel erarbeitet.

Um das Verständnis zu erleichtern, wurde die Aufgabe überarbeitet. Die Lernenden mussten nun lediglich die vorgegebenen Randbedingungen den entsprechenden Eigenschaften gesprochener Sprache zuordnen. Zusätzlich wurden Gegenbeispiele für den Graphen der Zuordnungsfunktion eingeführt (vgl. Abbildung 19.13). Diese Beispiele dienen als visuelle Unterstützung und sollen gleichzeitig dabei helfen, zu überprüfen, ob die Zuordnung der Randbedingungen nur zufällig korrekt war oder ob tatsächlich verstanden wurde, welche Bedeutung die Eigenschaften der gesprochenen Sprache für die Zuordnungsfunktion haben.

Die Zusatzaufgabe zur Bestimmung der Anzahl der verschiedenen Zuordnungen wurde nur einmal bearbeitet. Hierbei fiel auf, dass sehr viel Hilfestellung geleistet werden musste, weshalb zwei Tipps zu dieser Aufgabe hinzugefügt wurden, die ein strategischeres Suchen der Zuordnungspfade fördern sollen.

Außerdem wurde beobachtet, dass das Vervollständigen der Tabelle für die Zwischenspeicherung der Abstandswerte und damit das Ausführen des DTW-Verfahrens schwer fiel. Es wurde deshalb ein Tipp hinzugefügt, in welchem die Berechnung zweier weiterer Abstandswerte exemplarisch beschrieben wurde.

In der Abschlussdiskussion war zu beobachten, dass die Lernenden sich rege in die Diskussionen einbrachten und diverse Meinung zu Nutzen, Problemen, Gefahren und persönlichen Erfahrungen äußerten. Als Probleme bei der Anwendung des entwickelten Spracherkennungssystems wurden wiederholt die schlechte Erkennungsleistung bei zu lauten Hintergrundgeräuschen, bei zu großer Entfernung bei der Aufnahme vom Mikrofon oder bei starkem Dialekt und undeutlicher Aussprache genannt. Auch die Begrenztheit des verfügbaren Vokabulars wurde als bedeutender Nachteil hervorgehoben.

Exemplarische Meinungen zu problematischen / nützlichen Anwendungen der Spracherkennung sowie ethisch bedenkliche Aspekte bei der Entwicklung von Spracherkennungssystemen waren:

- Die meisten bewerteten Spracherkennungssysteme als besonders nützlich für den eigenen Alltag. Sie äußerten jedoch Bedenken hinsichtlich potenzieller Missbrauchsmöglichkeiten, wie zum Beispiel die Möglichkeit, solche Systeme im großen Stil zum heimlichen Abhören von Personen und zum Durchsuchen von Gesprächen nach bestimmten Schlüsselwörtern zu verwenden.
- Im Hinblick auf die Entwicklung solcher Systeme wiesen die Lernenden besonders auf die Problematik unausgeglichenerer Testdaten hin. Sie verwiesen dabei auf den während der Erarbeitung der Qualitätsmaße erkannten Fehlschluss: eine hohe zugewiesene Genauigkeit des Spracherkennungssystems, obwohl es alle Sprachaufnahmen fälschlicherweise als dasselbe Wort klassifizierte. Die Ursache hierfür war die Verwendung unausgeglichenerer Testdaten. Es wurde kritisch angemerkt, dass Entwickler diese Strategie nutzen könnten, um die Leistungsfähigkeit ihres Spracherkennungssystems besser darzustellen. Von den Betreu-

enden wurde zusätzlich der Aspekt hinzugefügt, dass auch unausgeglichene Trainingsdaten problematisch sein können.²

Selbsteinschätzung der Lernenden laut der schriftlichen Befragung

Abschließend werden die Ergebnisse der schriftlichen Befragung zusammengefasst. Eine detaillierte Darstellung der Evaluationsergebnisse findet sich in Anhang C.1. Es ist zu beachten, dass nicht alle an der Umfrage teilgenommen haben und im Fall der Teilnahme auch nicht zwingend alle Fragen beantwortet wurden.

19 von 21 Lernenden gaben auf die Frage, ob sie durch den Workshop das mathematische Modellieren besser begriffen haben, die Antwort stimmt vollkommen oder stimmt eher an. Beim Lernzuwachs wurde außerdem vier Mal die mathematische Modellierung beziehungsweise der Modellierungskreislauf genannt. Dies deutet darauf hin, dass der Workshop einen Beitrag dazu leistet, die mathematische Modellierung als übergeordnetes Konzept für den Umgang mit realen Problemstellungen besser zu verstehen.

Auch wurde wiederholt hervorgehoben, dass der Workshop die Bedeutung von Mathematik für den Alltag erfahrbar gemacht habe, was eines der Teilziele der entwickelten Lernmaterialien darstellt (vgl. Abschnitt 4.1). Die Antworten der Teilnehmenden auf die Frage „Was hast du für dich persönlich durch die Teilnahme am Workshop gelernt?“ unterstreichen dies:

- „Was man mit Mathe machen kann“
- „wofür man mathematische Modellierung alles benötigt“
- „Mathe ist nicht nur das, was wir im Unterricht machen“

Die allermeisten Lernenden (17 von 21) fanden, dass die Inhalte klar vermittelt wurden. Ein Teilnehmer/ eine Teilnehmende äußerte nach der ersten Durchführung, er/ sie habe angenommen, dass der Fokus noch stärker auf Künstlicher Intelligenz liege. Daraufhin wurde bei der Erarbeitung des Nächster-Nachbar-Verfahrens deutlicher herausgestellt, dass es sich dabei um Maschinelles Lernen und damit eine KI-Methode handelt, und die Präsentation wurde entsprechend angepasst.

Insgesamt wurde die Gestaltung der Arbeitsblätter offenbar als gut empfunden. 17 von 22 Personen stimmten der Aussage, dass die Gestaltung der Arbeitsblätter gut ist mit vollkommen oder eher zu. Zudem kommentierten die Lernenden „Ich fand den Vortrag und die Gestaltung gut.“ sowie „[...] gute Darstellung durch Präsentationen [...]“ und auf die Frage, was besonders gefallen hat, wurde die „allgemeine Gestaltung“ genannt.

²Eine Studie der University of Washington fand beispielsweise heraus, dass die Spracherkennungssoftware von Google Sprache von männlichen Sprechern sehr viel genauer als Sprache von weiblichen Sprechern erkennt (Tatman, 2017, S. 55). Als ein entscheidender Grund wird der unausgeglichene Trainingsdatensatz vermutet. In vorangegangenen Studien wurde beispielsweise aufgedeckt, dass bei vielen Spracherkennungssystemen typischerweise deutlich mehr als die Hälfte der Trainingsdaten von männlichen Sprechern stammen (Tatman, 2017, S. 57).

Die Aussage, dass die automatischen Rückmeldungen nach den Eingaben im Code hilfreich ist, bewertete die Mehrheit (18 von 22) mit stimmt vollkommen oder mit stimmt eher. Auch stimmte die Mehrheit (17 von 21) vollkommen oder eher zu, dass sie die Aufgaben eigenständig bearbeiten konnten. Das Anwenden einer Programmiersprache (auch wenn dies im Workshop nur an sehr wenigen Stellen mit angegebenen Befehlen erforderlich war) schien für Personen schwierig und für weitere zwei eher schwierig gewesen zu sein. Alle anderen stimmten der Aussage, dass ihnen der Umgang mit der Programmiersprache schwer fiel, allerdings gar nicht oder eher nicht zu.

Bezüglich der Hilfekarten zeichnete sich ein eher gemischtes Bild ab. Der Aussage, die Hilfekarten seien hilfreich, stimmten die Hälfte der Teilnehmenden eher oder vollkommen zu. 11 Lernende stimmten dieser Aussage nicht oder eher nicht zu. Dies kann auf die Tatsache zurückzuführen sein, dass bei der ersten Durchführung noch wenig Tippkarten zur Verfügung standen. Insbesondere auf dem sechsten Arbeitsblatt wurden daher weitere Hilfekarten hinzugefügt.

Die Plenums- und Arbeitsphasen sowie die Pausen scheinen zeitlich geeignet platziert worden zu sein. 10 Personen stimmten der Aussage, dass die Lern- und Arbeitszeit angemessen war, vollkommen zu und sieben stimmten ihr eher zu.

Die Schwierigkeit der Aufgaben scheint angemessen zu sein. Vier Lernende hielten die Aufgaben für zu einfach, allerdings befanden 17 diese Aussage eher nicht oder gar nicht zutreffend. Ähnlich fallen die Antworten auf die Aussage aus, dass die Aufgaben zu schwierig waren. 12 Lernende stimmten dem eher zu, acht stimmten eher nicht oder gar nicht zu und nur eine Person stimmte der Aussage vollkommen zu. Ein Teilnehmender / eine Teilnehmende gab an, dass ihm/ ihr besonders gefiel, dass „es ein relativ komplexes Thema war, aber dennoch gut machbar war“, was zum einen ebenfalls unterstreicht, dass der Schwierigkeitsgrad angemessen war und zum anderen zeigt, dass es gelungen ist, ein anspruchsvolles Thema didaktisch zu reduzieren.

18 von 21 Lernenden stimmten eher oder vollkommen zu, dass ihnen der Workshop insgesamt gut gefallen habe. Auf die Frage, ob ihnen etwas nicht gefallen habe, antworteten alle mit „Nein“ und dreiviertel der Lernenden stimmten vollkommen oder eher zu, dass sie den den Workshop weiterempfehlen würden. Dies spiegelt auch die Schulnote wider, die man dem Workshop geben sollte. Mehr als 63% bewerteten den Workshop mit einer zwei und 9% gaben dem Workshop sogar die Note eins.

20.2 Reflexion

In diesem Abschnitt wird das entwickelte Lernmaterial didaktisch reflektiert und dabei zusammenfassend erläutert, wie die in Kapitel 7 genannten Design-Prinzipien in diesem Lehr- und Lernmaterial umgesetzt werden.

Sprachbasierte Technologien wie etwa intelligente Sprachassistenten, automatische Auskunftssysteme, Diktier- und Transkriptionssysteme, automatische Übersetzungsdienste sind aus Industrie und Alltag kaum mehr wegzudenken. Schülerinnen und

Schüler erleben die Erkennung gesprochener Sprache besonders bei der Nutzung natürlichsprachlicher Assistenzsysteme wie Siri und Alexa, die in ihrem täglichen Leben eine zunehmend wichtigere Rolle spielen. Die Spracherkennung stellt daher ein konkretes Beispiel aus der Lebenswelt der Lernenden dar und kann als besonders relevante Problemstellung angesehen werden (vgl. Kapitel 7, DP5).

Zudem ist die Spracherkennung ein hochaktuelles Forschungsgebiet, in dem bereits in den 1960er-Jahren erste Fortschritte erzielt wurden, dessen Komplexität jedoch bis heute intensive Forschung erfordert (Kandji et al., 2024). Wissenschaftlerinnen und Wissenschaftler befassen sich weiterhin mit der zentralen Frage, die auch im Lernmaterial aufgegriffen wird: „Wie kann die Erkennung von gesprochener Sprache (noch besser) funktionieren?“. Die Fragestellung ist dadurch als authentisch einzustufen, da sie direkt an ein reales Forschungsproblem anknüpft (vgl. Kapitel 7, DP5).

Das in diesem Lernmaterial vorgestellte Modell kommt vorrangig bei automatischen Telefonabfragen oder Pick-to-Voice-Systemen in Lagerhäusern zum Einsatz, da sich die Methode des Dynamic Time Warping nur für die Erkennung von kurzen Befehlen eignet (Euler, 2006, S. 65). Für die Erkennung fließend gesprochener Sprache wird heutzutage vorrangig auf neuronale Netze als Klassifizierungsverfahren gesetzt (Kandji et al., 2024, S. 189). Die grundlegenden Schritte zur Entwicklung eines Spracherkennungssystems sind jedoch diejenigen, die auch im Lernmaterial vorgestellt werden. So sind die mathematischen Methoden, welche zur Vorverarbeitung (Fourier-Transformation und Mel-Transformation zur Erzeugung eines Sprachmusters) genutzt werden, auch in den eingangs genannten Sprachverarbeitungstechnologien grundlegend (Euler, 2006, S. 29 - 45; Yalova et al., 2023, S. 141 - 142). Damit ist auch die Verwendung der mathematischen Methode als authentisch zu bezeichnen (vgl. Kapitel 7, DP5). Bei der Wahl der Methode steht außerdem im Vordergrund, das Modell so einfach wie möglich und zugleich so komplex wie nötig zu gestalten. Das Ziel ist es, Schülerinnen und Schülern die Inhalte an einem Modellierungstag verständlich zu vermitteln und gleichzeitig ein funktionsfähiges System zu entwickeln, das Sprachsignale – wenn auch nur aus einer begrenzten Auswahl von Wörtern – zuverlässig erkennt. Die Erfahrung, gemeinsam ein funktionierendes System zu erschaffen, wurde von den Lernenden nach der Durchführung häufig als motivierend hervorgehoben und die getroffene Entscheidung damit nachträglich bestätigt.

Die im Lernmaterial genutzten Strategien und Methoden aus dem ML sind über die konkrete Problemstellung hinaus für zahlreiche weitere Anwendungen grundlegend. Dazu zählen die Teilung des Datensatzes in Test- und Trainingsdaten, das Nächster-Nachbar-Verfahren sowie die Darstellung der Testphase mithilfe einer Wahrheitstabelle und die Ableitung von Qualitätsmaßen aus der Wahrheitstabelle. Diese Strategien und Methoden werden von Hazzan und Mike (2023, S. 209) zu den zentralen Konzepten des Maschinellen Lernens gezählt.

Die als Trainings- und Testdaten genutzten Sprachaufnahmen sind reale Audioaufnahmen englischer Zahlenwörter, die von verschiedenen Sprechern eingesprochen wurden (Warden, 2018). Dies trägt zwar zur Komplexität der aus den Sprachsignalen erzeugten Spektrogramme bei, ermöglicht jedoch die realitätsnahe Funktionsweise des Spracherkennungssystems und es ist anzunehmen, dass dies die Motivation

der Lernenden positiv beeinflusst. Die Verwendung dieser realen Audiodaten kann daher als authentisch und praxisnah angesehen werden. Insgesamt erweist sich das betrachtete Modellierungsproblem als relevant und authentisch im Hinblick auf die Fragestellung und die Verwendung der mathematischen Methode und der Daten (vgl. Kapitel 7, DP5).

Wie bereits in Abschnitt 19.3 deutlich dargelegt, ist das Lernmaterial mathematisch reichhaltig und ermöglicht es, Inhalte aus verschiedenen mathematischen Bereichen zu thematisieren und zu verknüpfen (vgl. Kapitel 7, DP3). Aus dem Bereich der Analysis spielen beispielsweise trigonometrische Funktionen bei der Beschreibung von Tönen eine wichtige Rolle, aus dem Bereich der Geometrie ist bei der Klassifizierung der Abstand zwischen Vektoren als Kriterium ausschlaggebend, und bei der Bewertung des Modells werden stochastische Inhalte wie die Vierfeldertafel und relative Häufigkeiten thematisiert.

Neben den mathematischen Inhalten, die in diesem Lernmaterial im Fokus stehen, sind bei der Modellierung von Sprachsignalen physikalische Grundlagen aus der Signalverarbeitung von zentraler Bedeutung. Das algorithmische Arbeiten, wie beispielsweise bei der Bestimmung des optimalen Zuordnungspfades zweier Sprachmuster mit dynamischer Programmierung, hebt zudem die Vernetzung zur Informatik hervor. Das Lernmaterial kann somit nicht nur innermathematisch, sondern auch außermathematisch als interdisziplinär bezeichnet werden (vgl. Kapitel 7, DP3).

Durch das wiederholte Einordnen der in der vorangegangenen Phase erfolgten Tätigkeit in den vierschrittigen Modellierungskreislauf wird zudem transparent, welche Schritte beim Modellieren idealtypisch durchlaufen werden. Die Entwicklung eines Spracherkennungssystems erlaubt das vollständige und mehrfache Durchlaufen des Modellierungskreislaufs. Dabei wird mit einem möglichst einfachen Modell gestartet, mit dessen Hilfe zuerst nur einzelne Töne, anschließend Dreiklänge und schließlich Sprachsignale erkannt werden sollen. Der Lösungsfortschritt verbessert sich dabei mit jedem Durchlauf durch den Modellierungskreislauf. Gleichzeitig wird jedoch klar aufgezeigt, dass immer Vereinfachungen der realen Situation vorgenommen werden müssen, um das Problem lösen zu können.

Die Validierung der mathematischen Resultate ist ein Schritt der im Lernmaterial ausführlich behandelt wird (vgl. Abschnitt 19.2.5), da die Forschung gezeigt hat, dass viele Lernende Schwierigkeiten mit diesem Modellierungsschritt haben und deshalb mehr Übung benötigen (Maaß, 2004, S. 161). Mathematische Inhalte wie beispielsweise die Qualitätsmaße, das Nächster-Nachbar-Verfahren oder der Vergleich von Zeitreihen werden für die Lösung des Modellierungsproblems problembezogen eingeführt (vgl. Kapitel 7, DP4).

Ziel dieser Arbeit ist außerdem, zu erarbeiten, wie die Funktionsweise von KI-Systemen aus dem Kontext Verarbeitung natürlicher Sprache vermittelt werden und die diesbezügliche Rolle der Mathematik mit den Lernenden herausgearbeitet werden kann. Zunächst wird im Lernmaterial das Spracherkennungssystem als KI-System eingeordnet. Das System übernimmt Aufgaben, die Intelligenz erfordern, wenn sie vom Menschen übernommen werden. Insgesamt können durch die Bearbeitung des Lernmaterials alle zentralen Kompetenzen, welche die Funktionsweise

von KI-Systemen betreffen und in Kapitel 6 als übergreifende Lernziele formuliert werden, fördern (vgl. Abschnitt 19.1). Bei der Wahl der mathematischen Methoden wird darauf geachtet, dass sie möglichst einfach zugänglich sind, allerdings auch zu einem funktionsfähigen System führen (vgl. Kapitel 7, DP4).

Das Nächster-Nachbar-Verfahren wird als Klassifikationsverfahren ausgewählt, da es nur geringe mathematische Vorkenntnisse erfordert, eine intuitiv nachvollziehbare Funktionsweise aufweist und zugleich eine geeignete Grundlage bietet, um viele grundlegende Konzepte des Maschinellen Lernens zu veranschaulichen (Hazzan & Mike, 2022, S. 20 – 21). Die Herausforderung, dass die Klassifizierung bei der Spracherkennung mit mehrdimensionalen Sprachmerkmalen erfolgt, wird durch eine schrittweise Herangehensweise adressiert. Zunächst werden die Sprachmerkmale auf zwei- und später auf drei Dimensionen reduziert, um die Methode im zwei- und dreidimensionalen Anschauungsraum leichter zugänglich zu machen. Die anschließende Erweiterung auf mehrere Dimensionen ermöglicht es, die mehrdimensionale Klassifikation schrittweise verständlich zu vermitteln und dabei die Komplexität des Verfahrens sukzessive zu erhöhen.

Die Datenaufbereitung stellt eine weitere Herausforderung dar, da sie schnell sehr komplex werden kann. Um ein funktionsfähiges System zu entwickeln, ohne die Lernenden zu überfordern, werden nur die notwendigsten Schritte zur Vorverarbeitung vorgenommen. Die Fouriertransformation wird aufgrund ihrer Komplexität als Grey Box behandelt. Die grundlegende Idee der gefensterten Fouriertransformation wird zwar behandelt, jedoch kommt sie im Lernmaterial nur zur Anwendung. Die Möglichkeit, die Fourierkoeffizienten zu bestimmen, ist lediglich in einem Zusatzmaterial enthalten. Die weitere Aufbereitung der Daten erfolgt anschließend durch die Filterung mit einer Mel-Filterbank. Diese Methode kann den Lernenden als Mittelwertbildung mit unterschiedlichen breiten Frequenzblöcken zugänglich gemacht werden. Auch das Verfahren des Dynamic Time Warpings kann durch stärker angeleitete Aufgabenstellungen zugänglich gemacht werden. Insbesondere hilfreich hier ist die Dimensionsreduktion der betrachteten Zeitreihen auf eindimensionale Zeitreihen, um das Verfahren mit den Lernenden zu erarbeiten. Es werden Hands-On-Tasks gestellt, in denen zum Beispiel das Verfahren des Dynamic Time Warpings konkret an zwei beispielhaften eindimensionalen Sprachmustern durchgeführt werden sollen. Unter Hands-On-Tasks werden nach Hazzan und Mike (2022, S. 42, 235 – 236) Aufgaben verstanden, bei denen Lernende die Berechnungen des Algorithmus an einfachen Beispielen selbstständig durchführen bevor sie für komplexere Beispiele vom digitalen Werkzeug übernommen werden. Das soll helfen, den Algorithmus nachzuvollziehen und ihn als Prozess zu verstehen (Hazzan & Mike, 2022, S. 42).

Jupyter Notebooks werden in der Lerneinheit so eingesetzt, dass sie beim Lösen des Modellierungsproblems bestmöglich unterstützen (vgl. Kapitel 7, DP1). Aufwändige Berechnungen, die wenig zur mathematischen Einsicht beitragen, werden durch das digitale Tool effizient ausgelagert. So wird beispielsweise das Dynamic Time Warping nur in einer beispielhaften eindimensionalen Anwendung für Sprachmuster betrachtet (vgl. Abschnitt 19.2.6), da sich das Verfahren für mehrdimensionale Zeitreihen strukturell kaum verändert. Auf diese Weise bleibt das Verfahren anschaulich und damit leichter verständlich und rechnerisch handhabbar, wodurch wertvolle

Zeit gespart wird. Zudem erfolgt das Überprüfen der mathematischen Resultate insbesondere im ersten und zweiten Modellierungszyklus mithilfe des digitalen Tools graphisch, sodass die Lernenden sich selbst korrigieren können. Insgesamt erlaubt das digitale Werkzeug vielfältige Visualisierungen und Animationen, die den Lernprozess insbesondere auf den Arbeitsblätter 1, 2, 3 und 6 unterstützen.

Ergänzend zu den übergreifenden methodischen Maßnahmen, die die Kommunikationsfähigkeit fördern sollen (vgl. Kapitel 9), enthält das Lernmaterial zahlreiche Aufgaben, die zur mathematischen Kommunikation und Reflexion anregen. So sollen beispielsweise Ergebnisse von Rechnungen im Sachkontext interpretiert, die Bedeutung verschiedener Qualitätsmaße erläutert und eine fundierte Entscheidung für eines der beiden entwickelten Spracherkennungssysteme getroffen werden. Dies betrifft insbesondere Arbeitsblatt 5 und 6, in denen die Modellergebnisse interpretiert und validiert werden sollen. Diese Aufgaben sollen die Schülerinnen und Schüler zum Kommunizieren über Mathematik anregen (vgl. Kapitel 7, DP2).

Das entwickelte Lernmaterial bietet somit einen praxisnahen Einblick in die Spracherkennung mit einem Fokus auf die relevanten mathematischen Inhalten, wobei sowohl authentische Daten als auch etablierte Methoden aus der Forschung und unter anderem auch Methoden des Maschinellen Lernens zum Einsatz kommen. Dadurch wird nicht nur das Verständnis für die Funktionsweise von KI-Systemen gefördert, sondern auch die Bedeutung mathematischer Konzepte in diesem Kontext herausgestellt.

21 Ausblick

In diesem Ausblick soll ein Blick auf mögliche Weiterentwicklungen des Lehr- und Lernmaterials zur Spracherkennung geworfen werden. Zunächst soll der Fokus auf weiterführende Forschungsaspekte im methodisch-didaktischen Bereich gelegt werden.

Ein häufiger Diskussionspunkt unter Lehrkräften nach der Durchführung des Workshops war der eher angeleitete Charakter des Modellierungsprozesses. Die Lernumgebung wird einerseits dafür gelobt, dass sie den komplexen Modellierungsprozess verständlich und schrittweise aufbereitet. Besonders positiv hervorgehoben wurde, dass es den Lehrkräften Sicherheit bietet, dass abgesehen von Freitextantworten für die meisten Aufgaben klare Musterlösungen bereitgestellt werden können. Andererseits wurde kritisiert, dass die Lernenden durch die stark vorstrukturierten Aufgaben weniger Freiheit haben, eigene Strategien zu erproben. Dies wird teilweise durch Plenumsphasen aufgefangen, in denen die eigenen Ideen gesammelt und besprochen werden, auch wenn letztlich nur eine dieser Ideen umgesetzt wird. Vor allem im ersten Teil des Materials könnten jedoch die Aufgabenstellungen weiter geöffnet werden. So wäre beispielsweise auf dem ersten Arbeitsblatt ein explorativerer Zugang zum Zusammenhang zwischen Tonhöhe und Lautstärke mit den Parametern der Sinusfunktion denkbar.

Eine weitere Überlegung betrifft die Anpassung des Materials an die organisatorischen Rahmenbedingungen des Schulunterrichts, indem es auf eine Doppelstunde komprimiert wird. Dabei muss jedoch klar sein, dass nicht mehr alle Schritte des Modellierungskreislaufs durchlaufen werden können. Eine mögliche Alternative wäre, sich auf bestimmte Schritte wie etwa die Merkmalsreduktion oder das Klassifizierungsverfahren und dessen Evaluation zu konzentrieren, während das Frequenzspektrum bereits als gegeben angenommen wird. Auch die Verbesserung des Modells müsste aus Zeitgründen unberücksichtigt bleiben.

Aus mathematisch-inhaltlicher Sicht ergeben sich ebenfalls interessante anknüpfende oder alternative Ansätze, die in das entwickelte Modell integriert werden könnten. Einige dieser Ansätze, die in der Forschung zur Spracherkennung bereits Anwendung finden, sollen hier exemplarisch genannt werden. Alle vorgeschlagenen Alternativen sowie deren Vergleich mit den bisher verwendeten Methoden bieten eine gute Gelegenheit zur Vertiefung und Diskussion der Thematik. Sie könnten im Lernmaterial in einem Informations- oder Zusatzblatt aufgearbeitet werden.

Wie in Abschnitt 17.1.2 beschrieben, wird derzeit das Hamming-Fenster verwendet, um den Leck-Effekt zu minimieren, ohne jedoch detailliert auf die Wahl dieser Fensterfunktion einzugehen. Die Betrachtung alternativer Festerfunktionen wie beispielsweise das Rechteck-Fenster oder das ebenfalls in der Sprachverarbeitung häufig

verwendete Hanning-Fenster (Pfister & Kaufmann, 2017, S. 66) und deren Vergleich bieten die Möglichkeit, den Effekt der Fensterfunktion auf das durch die Kurzzeitspektralanalyse erzeugte Frequenzspektrum genauer zu untersuchen.

Darüber hinaus wäre es möglich, mit den Lernenden die Funktionsweise der Fouriertransformation intensiver zu erarbeiten. Derzeit wird diese im Material als Grey Box behandelt und lediglich als Methode zur Darstellung von Signalen als Überlagerung von Sinusfunktionen eingeführt. Wohak (2021) zeigt jedoch, wie die mathematische Funktionsweise der Fouriertransformation durch die Darstellung als Matrix-Vektor-Multiplikation auch für Oberstufenschüler verständlich gemacht werden kann. Bereits vorhandenes digitales Lernmaterial zur Singulärwertzerlegung, das am Beispiel der Audiokomprimierung von Wohak entwickelt wurde, könnte hier integriert werden. Diese Aufgaben zum Erarbeiten der Fouriertransformation sind als Alternative zum aktuellen zweiten Arbeitsblatt denkbar und könnten besonders schnellen und interessierten Lernenden zur Verfügung gestellt oder als Erweiterung des Modellierungstages angeboten werden.

Als Klassifizierungsverfahren wird hier das Nächster-Nachbar-Verfahren verwendet (vgl. Abschnitt 17.2). Die Variante des Verfahrens, bei der anstelle eines Nachbarn die k nächsten Nachbarn berücksichtigt werden, wurde zwar vorgestellt, jedoch für den Anwendungsfall der Spracherkennung nicht genauer untersucht. Der Vorteil des k -nächste-Nachbarn-Verfahrens gegenüber der Berücksichtigung lediglich des nächsten Nachbarn liegt in der geringeren Fehleranfälligkeit bei Ausreißern einzelner Klassen (Fellbaum, 2012, S. 272 - 273). Eine Untersuchung der Auswirkung der Anzahl der betrachteten Nachbarn auf die Qualität des Spracherkennungssystems gibt Aufschluss über die optimale Wahl des Parameters k und bietet gleichzeitig die Gelegenheit, Generalisierungsaspekte von maschinellen Lernverfahren zu untersuchen. Nachteilig ist jedoch der erhöhte Rechenaufwand durch zahlreiche Abstandsberechnungen. Dieser lässt sich verringern, indem für jede Klasse ein Mittelpunkt- oder Schwerpunktvektor ermittelt und für die Distanzbestimmung zum unbekannten Merkmalsvektor verwendet wird (Fellbaum, 2012, S. 273). Die Klassenauswahl erfolgt dann wieder anhand des minimalen Abstands. Steht mehr Zeit zur Verfügung so ist es denkbar, die Lernenden das k -nächste-Nachbarn-Verfahren eigenständig erarbeiten zu lassen, wie etwa in einem von Schönbrodt (2022) beschriebenen Beispiel geschehen, bei dem Lernende ein Empfehlungssystem für Filme entwickeln.

Darüber hinaus lassen sich alternative Randbedingungen zur dynamischen Zeitanpassung als die in Abschnitt 17.3.1 beschriebenen heranziehen und deren Einfluss auf die Leistung des Spracherkennungssystems untersuchen. Gängige Ansätze erlauben beispielsweise, dass zwar keine gesamten Laute, aber eine geringe Anzahl von Merkmalsvektoren eines Musters übersprungen werden darf (Pfister & Kaufmann, 2017, S. 358). Die Kontinuitätsbedingung wird somit im Vergleich zur Formulierung in Abschnitt 17.14 gelockert. Darüber hinaus können andere Randbedingungen als die in Abschnitt 17.3.1 vorgestellten für die dynamische Zeitanpassung verwendet und ihre Auswirkung auf die Performance des Spracherkennungssystem untersucht werden. Gebräuchlich sind beispielsweise Randbedingungen, bei denen die Kontinuitätsbedingung insofern verstanden wird, dass zwar keine ganzen Laute, aber doch eine kleine Anzahl an Merkmalsvektoren eines Musters übersprungen werden dürfen

(Pfister & Kaufmann, 2017, S. 358). Ebenfalls üblich sind unterschiedliche Gewichtungen der einzelnen Pfaderweiterungen (Pfister & Kaufmann, 2017, S. 359 - 360; Müller, 2021, S. 146 - 147). Häufig werden diagonale Pfaderweiterungen mit einem Faktor größer 1 gewichtet. Damit wird sichergestellt, dass die Diagonale nicht bevorzugt wird, nur weil sie einen kürzeren Zuordnungspfad erzeugt und somit weniger Abstandswerte aufsummiert werden müssen (Pfister & Kaufmann, 2017, S. 359).

Teil V

Schlussbetrachtung

22 Zusammenfassung und Schlussfolgerungen

Das zentrale Entwicklungsziel dieser Arbeit war das Design verschiedener Lernumgebungen zum Thema Verarbeitung natürlicher Sprache für einen realitätsbezogenen Mathematikunterricht. Dieses Ziel wurde zunächst legitimiert, indem die Notwendigkeit der Behandlung realer Probleme im Mathematikunterricht herausgestellt sowie Gründe für die Betrachtung von KI und insbesondere NLP als Teilbereich von KI im Mathematikunterricht dargelegt wurden.

Die Formulierung übergeordneter Lernziele für die Modellierungstage mit KI-Inhalten diente der Konkretisierung des Entwicklungsziels. Die Lernziele orientieren sich am Kompetenzrahmen für KI von Long und Magerko (2020) und an den durch den Projektrahmen des Programmes CAMMP vorgegebenen Zielsetzungen. Sie sind den einzelnen Schritten des Maschinellen Lernens zugeordnet und beinhalten sowohl inhaltsbezogene als auch prozessbezogene Kompetenzen. Die Entwicklung der Lernumgebungen erfolgte schließlich unter Verwendung gezielt gewählter Design-Prinzipien, die das Erreichen der aufgestellten übergreifenden Lernziele unterstützen sollen. Diese Design-Prinzipien wurden basierend auf Prinzipien für einen adäquaten Statistikunterricht, um der datenlastigen Natur der Problemstellungen gerecht zu werden, sowie auf Kriterien guter Modellierungsaufgaben herausgearbeitet. Sie umfassen die Verwendung geeigneter technischer Hilfsmittel, die Förderung von Diskussion und Reflexion, die Integration von inner- und außermathematischen Vernetzungen, die Herausarbeitung der Rolle der Mathematik sowie die Berücksichtigung von Authentizität und Relevanz der Problemstellung. Zusätzlich wird durch Binnendifferenzierung auf die Heterogenität der Zielgruppe eingegangen. Des Weiteren wurde der ganzheitliche Ansatz der Modellierungsprojekte im Programm CAMMP dargelegt, die Verwendung von Jupyter Notebooks begründet und deren Ausgestaltung als Lernumgebung und Einsatz in den Modellierungsprojekten vorgestellt.

In den jeweiligen Abschnitten zu den entwickelten Lernumgebungen wurden die mathematischen Inhalte ausgewählter Methoden zur Lösung der beiden Probleme vorgestellt, didaktisch reduziert sowie zugehörige Aufgaben entwickelt und diese in einer kohärenten Lehr- und Lernumgebung umgesetzt. An vielen Stellen war es aufgrund der Vielzahl der mathematischen Inhalte lediglich möglich, einen kurzen Ausblick auf alternative Zugänge zu den im Lernmaterial behandelten zu geben. Dies eröffnet zweifellos Möglichkeiten, um darüber hinaus weitere Zugänge zu erforschen und zu vergleichen. Anschließend wurde das entwickelte Lernmaterial curricular eingeordnet und es wurde gezeigt, dass das Vorwissen der Lernenden intensiv genutzt werden kann, um darauf aufbauend diejenigen mathematischen Methoden zu entwickeln, die für die Entwicklung des jeweiligen KI-Systems vonnöten sind. Zusätzlich

wurden bisherige Erfahrungen aus den Durchführungen mit Lernenden beschrieben, und der Aufbau der Materialien wurde im Hinblick auf die Design-Prinzipien und Ziele des Lernmaterials reflektiert. In den abschließenden Ausblicken wurden mögliche Optimierungsansätze sowie mögliche zukünftige Arbeiten diskutiert.

Konkret wurden Lernumgebungen für einen Modellierungstag und eine Modellierungs-Doppelstunde zum Thema Wortvorhersage sowie für einen Modellierungstag zum Thema Spracherkennung entwickelt. Bei beiden Themen wurde die Problemstellung mit Methoden des überwachten Lernens angegangen. Dabei unterscheiden sich jedoch die zugrunde liegenden mathematischen Inhalte erheblich. Auch die didaktische Umsetzung, obwohl sie den Design-Prinzipien folgt, variiert in manchen Aspekten, was auf die Unterschiede in den behandelten Inhalten sowie deren Umfang zurückzuführen ist. So basiert das System zur Wortvorhersage auf einem probabilistischen Modell, das als Output eine Wahrscheinlichkeit liefert (*N*-Gramm-Modell), während im Lernmaterial zur Spracherkennung mit einem deterministischen Modell gearbeitet wird, das eine eindeutige Klassenzugehörigkeit bestimmt (Nächster-Nachbar-Verfahren). Darüber hinaus liegt der Schwerpunkt beim Lernmaterial zur Spracherkennung stärker auf der Datenvorverarbeitung, während die maschinelle Lernmethode selbst weniger Raum einnimmt. Im Gegensatz dazu wird im Lernmaterial zur Wortvorhersage die ML-Methode detaillierter betrachtet. Insgesamt lassen sich die Inhalte, die für die Entwicklung eines Sprachmodells zur Wortvorhersage behandelt werden müssen, sehr viel stärker eingrenzen. Der Kontext erlaubt daher im Vergleich eine tiefergehende Beschäftigung mit den mathematischen Inhalten, während der Kontext der Spracherkennung thematisch sehr breit gefächert ist und eine Behandlung vieler verschiedener Methoden notwendig ist, um ein Spracherkennungssystem entwickeln zu können. Bei einzelnen Methoden wie der Fouriertransformation konnte daher nur die Grundidee vermittelt werden. In beiden Fällen wurde deutlich, dass die Verarbeitung und Analyse von Sprachdaten aufgrund ihrer multidimensionalen beziehungsweise sequentiellen Struktur eine anspruchsvolle Aufgabe darstellt, die den Einsatz speziell darauf abgestimmter Methoden erfordert.

Für die Spracherkennung wurde vorgestellt, wie die Modellierung von Tönen als trigonometrische Funktionen sowie die Kurzzeitspektralanalyse über eine Unterteilung des Signals in einzelne Abschnitte und die darauffolgende Zerlegung des Signals in einzelne Sinusschwingungen mit Lernenden erarbeitet werden kann. Zudem wurde gezeigt, wie das Anwenden einer Filterbank als gewichtete Mittelwertsbildung zugänglich gemacht werden kann. Weitere mathematische Konzepte und Methoden wie der Vergleich von Zeitreihen, die Funktionsweise eines Abstandsklassifikators und in diesem Zusammenhang der euklidische Abstand im Mehrdimensionalen wurden untersucht und didaktisch aufbereitet. Abschließend macht dieser Teil der Arbeit Vorschläge für die Bewertung der Klassifikationsergebnisse und deren Interpretation, sowie die didaktische Reduktion dessen. Es konnte gezeigt werden, dass über die hier vorgestellten Methoden ein Spracherkennungssystem entwickelt werden kann, welches funktionsfähig ist und dessen Methoden trotzdem kognitiv zugänglich für Lernende sind. Dabei konnten alle Kompetenzen, die für das Verständnis von KI-Systemen als wichtig erachtet wurden, adressiert werden. Die Problemstellung hat

sich zudem als relevant und authentisch herausgestellt. Es zeigte sich, dass die Problemstellung inner- und außermathematische Vernetzungen ermöglicht.

Bei der didaktischen Analyse im Kontext der Wortvorhersage konnte zudem dargelegt werden, dass das *N*-Gramm-Modell aus mathematikdidaktischer Sicht eine geeignete ML-Methode darstellt, um sie mit Schülerinnen und Schülern zu thematisieren. Dies liegt einerseits an seiner niederschwelligen kognitiven Zugänglichkeit und der Nähe zu schulmathematischen Inhalten, andererseits an der Tatsache, dass am Beispiel des *N*-Gramm-Modells alle Kompetenzen adressiert werden können, die für die Funktionsweise von KI-Systemen als relevanten erachtet werden. In Verbindung mit einer Problemstellung, die sowohl relevant als auch authentisch ist und zugleich inner- und außermathematische Vernetzungen ermöglicht, konnte gezeigt werden, dass das Lernmaterial zur Wortvorhersage nicht nur den definierten Design-Prinzipien genügt, sondern auch die Zielsetzung der Arbeit erfüllt: einen realitätsbezogenen Unterricht zu gestalten, um die Relevanz der Mathematik aufzuzeigen und das Verständnis für KI und NLP zu fördern.

Der Vergleich der beiden Problemstellungen und der dabei angewandten Methoden zeigt auch, dass die Identifikation von grundlegenden Kompetenzen, die für ein KI-Verständnis im Kontext des NLP aus mathematischer Sicht notwendig sind, aufgrund der Unterschiedlichkeit der Methoden und Vorgehensweisen eine Herausforderung darstellt. Die formulierten übergreifenden Lernziele stellen einen ersten Versuch dar, zentrale Vorgehensweisen sowie die dabei erforderlichen Kompetenzen zu identifizieren und zu strukturieren. Dabei wurde deutlich, dass diese Lernziele für die betrachteten Problemstellungen und Methoden einen geeigneten Rahmen bieten konnten. Um die Übertragbarkeit der entwickelten Lernziele auf weitere ML-Problemstellungen und unterschiedliche methodische Ansätze zu prüfen und gleichzeitig zu klären, ob sie tatsächlich die aus mathematischer Sicht relevanten Kompetenzen für ein KI-Verständnis abbilden, wären weiterführende Untersuchungen erforderlich.

Insgesamt zeigt diese Arbeit am Beispiel der entwickelten Lernumgebungen zur Sprachverarbeitung auf, wie die didaktische Reduktion komplexer realer Probleme und der zur Lösung verwendeter (ML-)Methoden auf ein schulmathematisches Niveau gelingen kann. Das entwickelte Lernmaterial ermöglicht es, die mathematische Modellierung als Metakonzept für die Herangehensweise an reale Probleme kennenzulernen und die Rolle der Mathematik für das Lösen von Problemen aus dem Alltag der Schülerinnen und Schüler am Beispiel der Themen Wortvorhersage und Spracherkennung herauszuarbeiten. Zudem wird deutlich, dass Methoden des Maschinellen Lernens häufig neu eingekleidete, aber alt bekannte mathematische Methoden sind, deren zentralen Ideen bereits im Schulunterricht vermittelt werden können.

23 Ausblick

Neben den im Ausblick der einzelnen Kapitel zur Spracherkennung und Wortvorhersage thematisierten Fortführungen der konkreten Einzelthemen werden abschließend übergreifende und ergänzende Punkte kurz erläutert.

Die empirische Begleitforschung dieses Entwicklungsprojektes bestand in der Selbst-einschätzung der Lernenden mithilfe von Fragebögen nach der Durchführung der Modellierungsprojekte. Die Auswertung dieser Ergebnisse sowie die Beobachtungen während der Durchführungen helfen, eine Einschätzung hinsichtlich verschiedener Faktoren wie dem Schwierigkeitsgrad, dem Lernfortschritt, Hürden während des Lernprozesses sowie dem Interesse an der Problemstellung zu machen. Für eine fundierte Analyse der Lernwirksamkeit der Lernumgebung sind jedoch weitere empirische Untersuchungen notwendig. In Zukunft ist daher geplant, für eine konkretere Beurteilung der Lernwirksamkeit die Antworten der Lernenden auf den digitalen Arbeitsblättern zu erfassen und auszuwerten.

Parallel zu dieser Arbeit wurde von der Autorin und weiteren Forschenden zudem eine längsschnittliche quantitative Studie durchgeführt, welche die Motivationssteigerung im Bezug auf das Verständnis von KI durch die Modellierungstage mit KI-Bezug im Rahmen des Projektes CAMMP untersuchte (Geisler et al., [under review](#)). Zu den Workshops, welche KI-Bezug haben, zählen unter anderem, der Workshop zur Wortvorhersage und der Workshop zur Spracherkennung. Da im Untersuchungszeitraum der Workshop zur Wortvorhersage jedoch nicht durchgeführt wurde, konnte dieser nicht in die Studie einbezogen werden. Auf Grundlage der Erwartungs-Wert-Theorie wurde ein etablierter Fragebogen zu Wertüberzeugungen hinsichtlich KI adaptiert und sowohl zu Beginn als auch am Ende des jeweiligen Workshops durchgeführt. Die erhobenen Daten zeigen einen Anstieg des Selbstkonzepts der Lernenden bezüglich des Verständnisses von KI durch die Workshops. Eine Änderung der Überzeugung bezüglich der Nützlichkeit eines Verständnisses von KI war allerdings wider Erwarten nicht zu beobachten. Die Studienergebnisse liefern wertvolle Hinweise für die Weiterentwicklung des Lernmaterials. Die fehlende Steigerung der Nützlichkeit in Bezug auf das Verstehen von KI könnte möglicherweise darauf zurückzuführen sein, dass die Workshops zwar die technische Relevanz von KI betonten, jedoch keine ausreichende Verknüpfung zu den individuellen Zielen und Interessen der Lernenden herstellten (Geisler et al., [under review](#)). Es könnte helfen, die Nützlichkeit im Hinblick auf deren unterschiedliche Dimensionen, wie etwa deren Bedeutung für die Schullaufbahn, die Gesellschaft, das Berufsleben oder den Alltag, expliziter anzusprechen. Auch eine stärkere Fokussierung auf einen Transfer des Gelernten auf weitere Anwendungsbereiche von KI, insbesondere solche, die für die Lernenden relevant sind, könnten die Nützlichkeit eines Verständnisses von KI positiv beeinflussen.

Dies könnte für die Weiterentwicklung der Lernumgebungen bedeuten, dass mehr Zeit verwendet wird, um unterschiedliche Anwendungsszenarien von Spracherkennungssystemen zu sammeln und auch, dass die zentralen Schritte bei der Entwicklung von KI-Systemen am Beispiel eines weiteren KI-Systems umrissen werden.

Ein langfristiges Ziel ist zudem, dass Lehrkräfte die entwickelten Lernmaterialien vermehrt in ihrem eigenen Unterricht einsetzen. Um dies zu erleichtern, wurde das Material durch ausführliches Begleitmaterial ergänzt, um die Hürde zu senken, solche Modellierungsprojekte eigenständig im Unterricht durchzuführen. Darüber hinaus wurde das Lernmaterial in Fortbildungen für Lehrkräfte sowie in Seminaren für Studierende des Lehramtes vorgestellt, und von den angehenden Lehrkräften ausprobiert und diskutiert. Obwohl das Material bereits vereinzelt von Lehrkräften genutzt wird, zeigt sich, dass es überwiegend im Rahmen von Modellierungsveranstaltungen unter der Betreuung wissenschaftlicher und studentischer Mitarbeiter des CAMMP-Projekts zum Einsatz kommt. Die positiven Rückmeldungen nach diesen betreuten Durchführungen stehen im Gegensatz zur eher geringen Nutzung durch Lehrkräfte im eigenen Unterricht. Häufig geben Lehrkräfte Zeitknappheit und den Umgang mit einem für sie unbekannten digitalen Medium als Hindernisse an.

Dem Aspekt der Zeitknappheit wird versucht durch das Angebot von Lernmaterial für Doppelstunden zu begegnen. Die Veröffentlichung des Lernmaterials zur Wortvorhersage in einem praxisorientierten Journal (Hofmann & Frank, 2024) soll das Material bei Lehrkräften bekannt machen und ihnen einen besonders einfachen Zugang ermöglichen. Nutzungszahlen dieser Lernmaterialien sind nicht bekannt.

Der Umgang mit Jupyter Notebooks als neues digitales Tool wurde bisher nicht explizit adressiert. Studien zeigen, dass die Häufigkeit der Nutzung von Informations- und Kommunikationstechnologien stark mit der Selbstwirksamkeitserwartung der Lehrkräfte im Umgang mit digitalen Medien sowie deren Einstellung zu diesen Technologien zusammenhängt (Ostermann et al., 2021, S. 201; Prasse, 2012, S. 161 - 166). Eine gezielte Untersuchung des Fortbildungsbedarfs zum digitalen Medium und eine Ausgestaltung des Fortbildungsangebots zur computergestützten mathematischen Modellierung, etwa mithilfe von Jupyter Notebooks, könnte daher sinnvoll sein. Es könnte nicht nur die Selbstwirksamkeit der Lehrkräfte stärken, sondern auch ihre Einstellung zu digitalen Tools und ihre Bereitschaft, diese im Unterricht einzusetzen, positiv beeinflussen. Die Weiterentwicklung adäquater Fortbildungsangebote, aber auch die Erforschung allgemeiner Gelingensbedingungen für den Einsatz der Lehr- und Lernmaterialien im Unterricht sind daher als langfristiges Ziel anzusehen.

Bereits im jeweiligen Ausblick zu den einzelnen Kapiteln ist sichtbar geworden, dass sich weitere inhaltliche Anknüpfungspunkte zu den in den Lernmaterialien behandelten Themen anbieten, um sie mit Schülerinnen und Schülern im Mathematikunterricht zu thematisieren. Mit Blick auf die steigende Popularität des Themas KI und der daraus folgenden Dringlichkeit der Förderung eines KI-Verständnisses ist es sicherlich lohnend, weitere Methoden des Maschinellen Lernens im Hinblick auf ihre didaktische Reduzierbarkeit für den Mathematikunterricht zu untersuchen. Der Bereich der Verarbeitung natürlicher Sprache als relevanter Teilbereich der KI ist ein weites Feld, in dem sich neben den Themen Wortvorhersage und Spracherkennung weitere interessante und für den mathematischen Modellierungsunterricht

geeignete Problemstellungen verbergen. Als Beispiel sei hier das Konzept der Word Embeddings genannt, dessen mathematischer Inhalt sich für den Schulunterricht didaktisch aufbereiten ließe. Dabei handelt es sich um ein Verfahren zur Konstruktion eines Vektorraums, in dem die Beziehungen zwischen den Vektoren die semantische Ähnlichkeit der von ihnen repräsentierten Wörter widerspiegeln (Di Gennaro et al., 2021). Dieses Verfahren bildet die Grundlage vieler Anwendungen des NLP und bietet die Möglichkeit, an einem spannenden aktuellen Thema schulmathematische Inhalte wie Abstände zwischen Vektoren, Normierung oder die grafische Interpretation des Skalarproduktes zu thematisieren.

Literatur

- Aebli, H. (2006). *Zwölf Grundformen des Lehrens. Eine allgemeine Didaktik auf psychologischer Grundlage. Medien und Inhalte didaktischer Kommunikation, der Lernzyklus.* (13. Aufl.). Stuttgart: Klett-Cotta.
- Arnold, K. C., Chauncey, K., & Gajos, K. Z. (2018). Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. *Graphics interface*, 42–49.
- Bauer, S., Büchter, A., & Henn, H. W. (2023). Schulmathematik und Realität – Verstehen durch Anwenden. In R. Bruder, A. Büchter, H. Gasteiger, B. Schmidt-Thieme & H.-G. Weigand (Hrsg.), *Handbuch der Mathematikdidaktik* (S. 21–56). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-66604-3_2
- Bauer, S., & Lengnink, K. (2024). Bist du auf dem Arbeitsmarkt vermittelbar? Algorithmische Entscheidungsfindung hinterfragen. *mathematik lehren*, 2024 (244), 36–41.
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. <https://doi.org/10.1145/3442188.3445922>
- Ben-Zvi, D., Gravemeijer, K., & Ainley, J. (2018). Design of statistics learning environments. *International handbook of research in statistics education*, 473–502. <https://hal.science/hal-02410840/document>
- Biehler, R., Engel, J., & Frischemeier, D. (2023). Stochastik: Leitidee Daten und Zufall. In R. Bruder, A. Büchter, H. Gasteiger, B. Schmidt-Thieme & H.-G. Weigand (Hrsg.), *Handbuch der Mathematikdidaktik* (S. 243–278). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-66604-3_8
- Biehler, R., Schönbrodt, S., & Frank, M. (2024). KI als Thema für den Mathematikunterricht. *mathematik lehren*, 2024 (244), 2–7.
- Bishop, C. (2007, Oktober). *Pattern Recognition and Machine Learning (Information Science and Statistics)*.
- Blomhoej, M., & Højgaard, T. (2003). Developing mathematical modelling competence: Conceptual clarification and educational planning. *Teaching Mathematics and Its Applications*, 22, 123–139. <https://doi.org/10.1093/teamat/22.3.123>
- Blum, W. (1985). Anwendungsorientierter Mathematikunterricht in der didaktischen Diskussion. *Mathematische Semesterberichte*, 32(2), 195–232.
- Blum, W. (2007). Mathematisches Modellieren - zu schwer für Schüler und Lehrer? *Beiträge zum Mathematikunterricht*, 3–12.
- Blum, W. (2010). Modellierungsaufgaben im Mathematikunterricht. Herausforderung für Schüler und Lehrer. *Praxis der Mathematik*, 34(52), 42–48.

- Blum, W. (2015). Quality teaching of mathematical modelling: What do we know, what can we do? *The proceedings of the 12th international congress on mathematical education: Intellectual and attitudinal challenges*, 73–96.
- Blum, W., & Leiß, D. (2005). Modellieren im Unterricht mit der „Tanken“-Aufgabe. *mathematik lehren*, (128), 18–21.
- Blum, W., & Leiss, D. (2005). „Filling Up“—the problem of independence-preserving teacher interventions in lessons with demanding modelling tasks. *CERME 4—proceedings of the fourth congress of the European Society for Research in mathematics education*, 1623.
- Blum, W., & Niss, M. (1991). Applied mathematical problem solving, modelling, applications, and links to other subjects—State, trends and issues in mathematics instruction. *Educational studies in mathematics*, 22(1), 37–68.
- Botsch, B. (2023). *Maschinelles Lernen - Grundlagen und Anwendungen: Mit Beispielen in Python*. Springer Spektrum Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-67277-8>
- Brigola, R. (2012). *Fourier-Analysis und Distributionen: Eine Einführung mit Anwendungen*. Tredition.
- Bruner, J. S., & Hartung, A. (1973). *Der Prozeß der Erziehung* (3. Aufl.). Berlin Verlag.
- Buchkremer, R. (2020). Natural Language Processing in der KI. In R. Buchkremer, T. Heupel & O. Koch (Hrsg.), *Künstliche Intelligenz in Wirtschaft & Gesellschaft: Auswirkungen, Herausforderungen & Handlungsempfehlungen* (S. 29–45). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-29550-9_2
- Bundesregierung Deutschland. (2018). Strategie Künstliche Intelligenz der Bundesregierung. www.bundesregierung.de/resource/blob/997532/1550276/3f7d3c41c6e05695741273e78b8039f2/2018-11-15-ki-strategie-data.pdf
- Burkhardt, H. (1989). Mathematical modelling in the curriculum. *Applications and modelling in learning and teaching mathematics*, 1–11.
- Busse, A. (2005). Individual ways of dealing with the context of realistic tasks—first steps towards a typology. *ZDM*, 37(5), 354–360.
- Carstensen, K.-U., Ebert, C., Ebert, C., Jekat, S., Langer, H., & Klabunde, R. (2009, November). *Computerlinguistik und Sprachtechnologie*. Spektrum Akademischer Verlag.
- Casal-Otero, L., Catala, A., Fernández-Morante, C., Taboada, M., Cebreiro, B., & Barro, S. (2023). AI literacy in K-12: a systematic literature review. *International Journal of STEM Education*, 10(1), 29.
- Chen, S. F., & Goodman, J. (1998). An empirical study of smoothing techniques for language modeling. *Harvard Computer Science Group Technical Report TR-10-98*, 1–63. <http://nrs.harvard.edu/urn-3:HUL.InstRepos:25104739>
- Computer-Based Math. (o. D.). The Computational Thinking Process Poster [Zugegriffen: 22.01.2025]. <https://www.computationalthinking.org/helix>
- Cormen, T. (2001). Introduction to Algorithms TH Cormen, CE Leiserson, RL Rivest, and C. Stein, eds.
- Cover, T. M., & Thomas, J. A. (1991). Entropy, relative entropy and mutual information. In *Elements of information theory* (S. 12–49). John Wiley & Sons.

- Cunningham, P., & Delany, S. J. (2021). k-Nearest Neighbour Classifiers - A Tutorial. *ACM Computing Surveys*, 54(6), 1–25. <https://doi.org/10.1145/3459665>
- Dahmen, W., & Reusken, A. (2008). *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg.
- Dale, R., Di Eugenio, B., & Scott, D. (1998). Introduction to the special issue on natural language generation. *Computational Linguistics*, 24(3), 345–353.
- Dewan, A., & Touretzky, D. (2024). Markov Chain Demo [Zugegriffen: 30.09.2024]. <https://www.cs.cmu.edu/~dst/MarkovChainDemo/>
- Di Gennaro, G., Buonanno, A., & Palmieri, F. A. (2021). Considerations about learning Word2Vec. *The Journal of Supercomputing*, 1–16.
- Döbel, I., Leis, M., Molina Vogelsang, M., Neustroev, D., Petzka, H., Riemer, A., Rüping, S., Voss, A., Wegele, M., & Welz, J. (2018). Maschinelles Lernen. Eine Analyse zu Kompetenzen, Forschung und Anwendung. https://www.bigdata-ai.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/BMBF_Fraunhofer_ML-Ergebnisbericht_Gesamt.pdf
- Eichler, A., & Vogel, M. (2009). *Leitidee Daten und Zufall* (2. Aufl.). Springer Spektrum, Wiesbaden.
- Euler, S. (2006). *Grundkurs Spracherkennung: Vom Sprachsignal zum Dialog - Grundlagen und Anwendungen verstehen - Mit praktischen Übungen*. Vieweg + Teubner Verlag. <https://doi.org/10.1007/978-3-8348-9113-6>
- Evangelista, I., Blesio, G., & Benatti, E. (2018). Why Are We Not Teaching Machine Learning at High School? A Proposal. <https://doi.org/10.1109/WEEF-GEDC.2018.8629750>
- Fellbaum, K. (2012). *Sprachverarbeitung und Sprachübertragung*. <https://doi.org/10.1007/978-3-642-31503-9>
- Fink, G. A. (2003). *Mustererkennung mit Markov-Modellen*. <https://doi.org/10.1007/978-3-322-80065-7>
- Forschungsgruppe LIMAS. (1971). LIMAS-Korpus [Zugegriffen: 05.08.2021]. <https://www.korpora.org/Limas/>
- Frank, M., Roeckerath, C., & Schönbrodt, S. (2022). Einführung. In M. Frank & C. Roeckerath (Hrsg.), *Neue Materialien für einen realitätsbezogenen Mathematikunterricht 9: ISTRON-Schriftenreihe* (S. 1–6). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-63647-3_1
- Freudenthal, H. (1973). *Mathematik als pädagogische Aufgabe* (Bd. 2). Klett, Stuttgart.
- Frochte, J. (2021). *Maschinelles Lernen. Grundlagen und Algorithmen in Python* (3. Aufl.). Carl Hanser Verlag.
- Geisler, S., Menzel, M., Hofmann, S., Weygandt, B., Bauer, S., & Schönbrodt, S. (under review). Lassen sich Wertüberzeugungen und Selbstkonzept zu Künstlicher Intelligenz durch KI-Modellierungsworkshops fördern? *mathematica didactica*.
- Giorgino, T. (2009). Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software*, 31(7), 1–24. <https://doi.org/10.18637/jss.v031.i07>

- Goethe, J. W. v. (1774a). *Die Leiden des jungen Werthers. Bd. 1* [In: Deutsches Textarchiv]. Leipzig. https://www.deutsches-textarchiv.de/goethe_werther01_1774
- Goethe, J. W. v. (1774b). *Die Leiden des jungen Werthers. Bd. 2* [In: Deutsches Textarchiv]. Leipzig. https://www.deutsches-textarchiv.de/goethe_werther02_1774
- Goethe, J. W. v. (1796). *Wilhelm Meisters Lehrjahre. Bd. 4* [In: Deutsches Textarchiv]. Frankfurt (Main) u. a. https://www.deutsches-textarchiv.de/goethe_lehrjahre04_1796
- Goethe, J. W. v. (1809). *Die Wahlverwandtschaften. Bd. 1* [In: Deutsches Textarchiv]. Tübingen. https://www.deutsches-textarchiv.de/goethe_wahlverw01_1809
- Goodfellow, I., Bengio, Y., & Courville, A. (2018). *Deep Learning: das umfassende Handbuch : Grundlagen, aktuelle Verfahren und Algorithmen, neue Forschungsansätze* (1. Auflage). mitp.
- Greefrath, G., Kaiser, G., Blum, W., & Borromeo Ferri, R. (2013). Mathematisches Modellieren—eine Einführung in theoretische und didaktische Hintergründe. *Mathematisches Modellieren für Schule und Hochschule: Theoretische und didaktische Hintergründe*, 11–37.
- Greefrath, G., & Siller, H.-S. (2018). *Digitale Werkzeuge, Simulationen und mathematisches Modellieren*. https://doi.org/10.1007/978-3-658-21940-6_1
- Grigoleit, S. (2019). Das Fraunhofer-Institut für Naturwissenschaftlich-Technische Trendanalysen berichtet über neue Technologien - Natural Language Processing. www.int.fraunhofer.de/content/dam/int/de/documents/EST/EST-0419-Natural-Language-Processing.pdf
- Güll, R. (2010). Zur Statistik des deutschen Wortschatzes. *Statistisches Monatsheft Baden-Württemberg*, 52–54. https://www.statistik-bw.de/Service/Veroeff/Monatshefte/PDF/Beitrag10_09_11.pdf
- Hamming, R. W. (1989). *Digital filters* (Bd. 3). Prentice-Hall, New York.
- Haque, M. M., Habib, M. T., & Rahman, M. M. (2016). Automated word prediction in bangla language using stochastic language models. *arXiv preprint arXiv:1602.07803*. <https://doi.org/10.5121/ijfcst.2015.5607>
- Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., & Ramage, D. (2019). Federated Learning for Mobile Keyboard Prediction. <https://arxiv.org/abs/1811.03604>
- Hayes, P., & Ford, K. (1995). Turing test considered harmful. *IJCAI (1)*, 972–977.
- Hazzan, O., & Mike, K. (2022). Teaching core principles of machine learning with a simple machine learning algorithm: the case of the KNN algorithm in a high school introduction to data science course. *ACM Inroads*, 13, 18–25. <https://doi.org/10.1145/3514217>
- Hazzan, O., & Mike, K. (2023, Januar). *Guide to Teaching Data Science*. <https://doi.org/10.1007/978-3-031-24758-3>
- Heesen, B. (2023). *Künstliche Intelligenz und Machine Learning mit R*. Springer.
- Hefendehl-Hebeker, L. (2022). Was ist und was soll die „Stoffdidaktik“? https://www.math.kit.edu/didaktik/seite/stoffdidaktik/media/22_kit-didaktik-ws_hefendehl-hebeker_va.pdf

- Henze, N. (2023). *Stochastik für Einsteiger* (14. Aufl.). Springer Spektrum, Berlin.
- Henze, N. (2025). Maximum-Likelihood-Schätzung: Multinomialverteilung [Zugegriffen: 17.01.2025]. https://www.youtube.com/watch?v=ybyBC_gj_XU
- Hewitt, E. S., & Hewitt, R. E. (1979). The Gibbs-Wilbraham phenomenon: An episode in fourier analysis. *Archive for History of Exact Sciences*, 21, 129–160. <https://api.semanticscholar.org/CorpusID:119355426>
- Hischer, H., & Lambert, A. (2002). *Mathematikunterricht und neue Medien: Hintergründe und Begründungen in fachdidaktischer und fachübergreifender Sicht*. Franzbecker Hildesheim.
- Hoeffer, K. (2022). *Aktivitätserkennung auf dem Smartphone - Entwicklung von Unterrichtsmaterial für computergestützte mathematische Modellierungsprojekte* [Masterarbeit]. KIT.
- Hoffmann, R., & Wolff, M. (2014). *Intelligente Signalverarbeitung 1*. Springer Vieweg.
- Hofmann, S., & Frank, M. (2022). Teaching data science in school: Digital learning material on predictive text systems. In G. Bolondi & J. Hodgen (Hrsg.), *Twelfth Congress of the European Society for Research in Mathematics Education (CERME12)*. <https://hal.archives-ouvertes.fr/hal-03751829>
- Hofmann, S., & Frank, M. (2023). Maschinelles Lernen im Schulunterricht am Beispiel einer problemorientierten Lerneinheit zur Wortvorhersage. *Beiträge zum Mathematikunterricht 2022: 56. Jahrestagung der Gesellschaft für Didaktik der Mathematik*, 157–160. <http://dx.doi.org/10.17877/DE290R-23317>
- Hofmann, S., & Frank, M. (2024). Wortvorschläge beim Chatten: KI und natürliche Sprachverarbeitung im Stochastikunterricht. *mathematik lehren*, 2024(244), 24–29. <https://www.friedrich-verlag.de/friedrich-plus/sekundarstufe/mathematik/stochastik/wortvorschlage-beim-chatten-18374>
- Hofmann, S., & Schönbrodt, S. (2024). Projektkurs für Mädels: Mit Mathe und KI reale Probleme lösen. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 32(2), 124–126. <https://doi.org/doi:10.1515/dmvm-2024-0037>
- Hofmann, S., Schönbrodt, S., & Kindler, S. (2024). Projektkurs: Mädels machen MI(N)T – KI-Bildung und Mädchenförderung. *Beiträge zum Mathematikunterricht 2024: 57. Jahrestagung der Gesellschaft für Didaktik der Mathematik*, 1592. <http://dx.doi.org/10.17877/DE290R-24890>
- Jablonka, E., & Lengnink, K. (2022). Gerechte Algorithmen...? *mathematik lehren*, 2022(230), 10–14.
- Jain, B. (2019). Making the Dynamic Time Warping Distance Warping-Invariant. <https://arxiv.org/abs/1903.01454>
- Jiang, S., & Chen, Z. (2023). Application of dynamic time warping optimization algorithm in speech recognition of machine translation. *Heliyon*, 9(11), e21625. <https://doi.org/https://doi.org/10.1016/j.heliyon.2023.e21625>
- Jordan, A., Krauss, S., Löwen, K., Blum, W., Neubrand, M., Brunner, M., Kunter, M., & Baumert, J. (2008). Aufgaben im COACTIV-Projekt: Zeugnisse des kognitiven Aktivierungspotentials im deutschen Mathematikunterricht. *Journal für Mathematik-Didaktik*, 29, 83–107.

- Jurafsky, D., & Martin, J. H. (2023). *Speech and Language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>
- Kandji, A. K., Ba, C., & Ndiaye, S. (2024). State-of-the-Art Review on Recent Trends in Automatic Speech Recognition. In M. Masinde, S. Möbs & A. Bagula (Hrsg.), *Emerging Technologies for Developing Countries* (S. 185–203). Springer Nature Switzerland.
- Kim, K., Kwon, K., Ottenbreit-Leftwich, A., Bae, H., & Glazewski, K. (2023). Exploring middle school students' common naive conceptions of Artificial Intelligence concepts, and the evolution of these ideas. *Education and Information Technologies*, 28(8), 9827–9854.
- Kindler, S., Schönbrodt, S., & Frank, M. (2023). Von der Schulmathematik zu künstlichen neuronalen Netzen. *Beiträge zum Mathematikunterricht 2022*.
- Klafki, W. (1995). Zum Problem der Inhalte des Lehrens und Lernens in der Schule aus der Sicht kritisch-konstruktiver Didaktik. In S. Hopmann & K. Riquarts (Hrsg.), *Didaktik und/oder Curriculum. Grundprobleme einer international vergleichenden Didaktik* (S. 91–102). Beltz. <https://doi.org/10.25656/01:10001>
- Klafki, W., & Stöcker, H. (1985). Innere Differenzierung des Unterrichts. Fünfte Studie. *Klafki, Wolfgang: Neue Studien zur Bildungstheorie und Didaktik. Beiträge zur kritisch-konstruktiven Didaktik*. Weinheim: Beltz.
- Krüger, K., Sill, H., & Sikora, C. (2015). *Didaktik der Stochastik in der Sekundarstufe I*. Springer Berlin Heidelberg. <https://books.google.de/books?id=B0vgCQAAQBAJ>
- Krüger, K., Sill, H.-D., & Sikora, C. (2015). *Didaktik der Stochastik in der Sekundarstufe I*. Springer Spektrum, Berlin.
- Kubat, M., Matwin, S., et al. (1997). Addressing the curse of imbalanced training sets: one-sided selection. *Icml*, 97(1), 179.
- Kultusministerkonferenz. (2012). *Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife: (Beschluss der Kultusministerkonferenz vom 18.10.2012)*. Wolters Kulver: München.
- Kultusministerkonferenz. (2022). *Bildungsstandards für das Fach Mathematik Erster Schulabschluss (ESA) und Mittlerer Schulabschluss (MSA): (Beschluss der Kultusministerkonferenz vom 15.10.2004 und vom 04.12.2003, i.d.F. vom 23.06.2022)*. Wolters Kulver: München.
- Laupichler, M. C., Aster, A., Schirch, J., & Raupach, T. (2022). Artificial intelligence literacy in higher and adult education: A scoping literature review. *Computers and Education: Artificial Intelligence*, 3, 100101.
- Long, D., & Magerko, B. (2020). What is AI Literacy? Competencies and Design Considerations. *CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3313831.3376727>
- M, L. N., & Kopparapu, S. K. (2014). Choice of Mel Filter Bank in Computing MFCC of a Resampled Speech. *CoRR*, abs/1410.6903. <http://arxiv.org/abs/1410.6903>
- Maaß, K. (2004). Mathematisches Modellieren im Unterricht—Ergebnisse einer empirischen Studie. *Journal für Mathematik-Didaktik*, 25, 175–176.

- Maafß, K. (2007). *Mathematisches Modellieren - Aufgaben für die Sekundarstufe I*. Berlin: Cornelsen Scriptor.
- Mallat, S. (2009). *A Wavelet Tour of Signal Processing - The Sparse Way*. Academic Press.
- Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing* (2. Aufl.). MIT Press, Cambridge.
- Markov, A. A. (1913). Primer statisticheskogo issledovaniya nad tekstom “Evgeniya Onegina”, illyustriruyuschij svyaz’ispytanij v cep’. *Izvestiya Akademii Nauk*, 7, 153.
- Martneau, P.-F. (2009). Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 306–318. <https://doi.org/10.1109/TPAMI.2008.76>
- Martignon, L., Engel, J., & Erickson, T. (2022). A Transparent, Simple AI Tool for Constructing Efficient and Robust Fast and Frugal Trees for Classification Under Risk. <https://doi.org/10.52041/iase.icots11.T6G3>
- McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. <http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf>
- McFee, B., McVicar, M., Faronbi, D., Roman, I., Gover, M., Balke, S., Seyfarth, S., Malek, A., Raffel, C., Lostanlen, V., van Niekirk, B., Lee, D., Cwitkowitz, F., Zalkow, F., Nieto, O., Ellis, D., Mason, J., Lee, K., Steers, B., ... Pimenta, W. (2024). *librosa/librosa: 0.10.2.post1* (Version 0.10.2.post1). Zenodo. <https://doi.org/10.5281/zenodo.11192913>
- Mertala, P., & Fagerlund, J. (2023). Finnish 5th and 6th graders’ misconceptions about artificial intelligence. *International Journal of Child-Computer Interaction*, 39, 100630. <https://doi.org/10.1016/j.ijcci.2023.100630>
- Miao, F., & Shiohira, K. (2024). *AI competency framework for students*. UNESCO. <https://doi.org/10.54675/jkjb9835>
- Ministerium für Kultur, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Mathematik: (vom 23. März 2016 in der Fassung vom 29. Februar 2024 (V2))*. www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_M.V2.pdf. 2024.
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Informatik, Mathematik, Physik (IMP)*. 2016. [https://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_IMP.pdf](http://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_IMP.pdf)
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Musik*. www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_MUS.pdf. 2016.
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Naturwissenschaft und Technik (NwT)*. 2016. [https://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_NWTPROFIL.pdf](http://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_NWTPROFIL.pdf)

- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Mathematik*. www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_M.pdf. 2016, März.
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Informatik, Wahlfach in der Oberstufe (Schulversuch)*. 2020. https://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/BP2016BW_ALLG_GYM_INFWFO/bildungsplan_BP2016BW_ALLG_GYM_INFWFO.pdf
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Physik: (Überarbeitete Fassung vom 25. März 2022)*. 2022. [www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_PH.V2%20\(2022-03-25\).pdf](http://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_GYM_PH.V2%20(2022-03-25).pdf)
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. (2023a). Digitalisierung [Zugegriffen: 30.09.2024]. <https://km.baden-wuerttemberg.de/de/schule/digitalisierung>
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. (2023b). Schulreform geht in die Anhörungsphase [Zugegriffen: 18.11.2024]. <https://km.baden-wuerttemberg.de/de/service/pressemitteilung/pid/schulreform-geht-in-die-anhoerungsphase?highlight=G9>
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. *Bildungsplan des Gymnasiums - Deutsch: (vom 23. März 2016 in der Fassung vom 29. Februar 2024 (V2))*. 2024. https://www.bildungsplaene-bw.de/site/bildungsplan/get/documents/lsbw/export-pdf/depot-pdf/ALLG/BP2016BW_ALLG_SEK1_D.V2.pdf
- Ministerium für Kultus, Jugend und Sport Baden-Württemberg. (2024b). Fragen und Antworten zur Bildungsreform [Zugegriffen: 18.11.2024]. <https://km.baden-wuerttemberg.de/de/schule/schulartuebergreifend/faq-bildungsreform>
- Müller, M. (2021, April). *Fundamentals of Music Processing* (2. Aufl.). Springer Cham. <https://doi.org/10.1007/978-3-030-69808-9>
- Myers, C. S., Rabiner, L. R., & Rosenberg, A. E. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28, 623–635. <https://api.semanticscholar.org/CorpusID:14297130>
- Myers, D. G. (2014). Intelligenz. In *Psychologie* (3. Aufl., S. 399–436). Springer Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-40782-6>
- Nadas, A. (1984). Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(4), 859–861.
- Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5), 544–551. <https://doi.org/10.1136/amiajnl-2011-000464>
- Nandi, A., & Jagadish, H. V. (2007). Effective phrase prediction. *Proceedings of the 33rd International Conference on Very Large Data Bases*, 219–230.

- NDR. (2024). Das Jugendwort 2023 steht fest: Es ist "goofy" [Zugegriffen: 16.07.2024]. <https://www.ndr.de/kultur/buch/Das-Jugendwort-2023-steht-fest-Es-ist-goofy.jugendwort234.html>
- Nebel, B. (2019). Turing-Test. In K. Liggieri & O. Müller (Hrsg.), *Mensch-Maschine-Interaktion: Handbuch zu Geschichte – Kultur – Ethik* (S. 304–306). J.B. Metzler. https://doi.org/10.1007/978-3-476-05604-7_57
- Neglectos. (2018). A preliminary analysis on the use of python notebooks [Zugegriffen: 11.09.2024]. <https://blog.bitergia.com/2018/04/02/a-preliminary-analysis-on-the-use-of-python-notebooks/>
- Ney, H., Martin, S., & Wessel, F. (1997). Statistical Language Modeling Using Leaving-One-Out. In S. Young & G. Blothoof (Hrsg.), *Corpus-Based Methods in Language and Speech Processing* (S. 174–207). Springer Netherlands. https://doi.org/10.1007/978-94-017-1183-8_6
- Niss, M. (1992). Applications and modeling in school mathematics – directions for future development. In I. Wirzup & R. Streit (Hrsg.), *Developments in school mathematics education around the world* (S. 346–361, Bd. 3). Reston: NCTM.
- Nugues, P. M. (2006). *An introduction to language processing with Perl and Prolog*. Springer.
- Oldenburg, R. (2008). Phrasendreschmaschine und Text-Evolution: Unterrichtsideen für Zeichenketten mit PYTHON. *LOG IN*, 154/155, 91–98.
- Opel, S., Schlichtig, M., Schulte, C., Biehler, R., Frischmeier, D., Podworny, S., & Wassong, T. (2019). Entwicklung und Reflexion einer Unterrichtssequenz zum Maschinellen Lernen als Aspekt von Data Science in der Sekundarstufe II. <https://doi.org/10.18420/infos2019-c14>
- Ortlieb, C. P. (2004). Mathematische Modelle und Naturerkenntnis. *Mathematica didactica*, 27(1), 23–40.
- Ostermann, A., Lindmeier, A., Härtig, H., Kampschulte, L., Ropohl, M., & Schwanewedel, J. (2021). Mathematikspezifische Medien nutzen: Was macht den Unterschied – Lehrkraft, Schulkultur oder Technik? *DDS – Die Deutsche Schule*, 113(2), 199–217.
- Palm, T. (2009). Theory of authentic task situations. In L. Verschaffel, B. Greer, W. Dooren & S. Mukhopadhyay (Hrsg.), *Words and Worlds: Modeling Verbal Descriptions of Situations* (S. 3–19). <https://doi.org/10.1163/9789087909383>
- Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.
- Pečarić, J., Perić, I., & Perić, J. (2015). *Combinatorial Improvements of Jensen's Inequality* (Bd. 8). Element.
- Pfister, B., & Kaufmann, T. (2017). *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Berlin Heidelberg. <https://books.google.de/books?id=g1YoDwAAQBAJ>
- Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering*, 26(65). <https://doi.org/https://doi.org/10.1007/s10664-021-09961-9>

- Pohjolainen, S., & Heiliö, M. (2016). Introduction. In S. Pohjolainen (Hrsg.), *Mathematical Modelling* (S. 1–5). Springer International Publishing. https://doi.org/10.1007/978-3-319-27836-0_1
- Pollak, H. O. (1977). The interaction between mathematics and other school subjects. *Proceedings of the Third International Congress on Mathematical Education*, 255–264.
- Prasse, D. (2012). *Bedingungen innovativen Handelns in Schulen. Funktion und Interaktion von Innovationsbereitschaft, Innovationsklima und Akteursnetzwerken am Beispiel der IKT-Integration an Schulen* [Diss.]. Humboldt-Universität.
- Prediger, S., Link, M., Hinz, R., Hußmann, S., Thiele, J., & Ralle, B. (2012). Lehr-Lernprozesse initiieren und erforschen–fachdidaktische Entwicklungsfor- schung im Dortmunder Modell. *Der mathematische und naturwissenschaftliche Unterricht*, 65(8), 452–457.
- Radev, D. (2013). *Puzzles in Logic, Languages and Computation: The Red Book*. <https://doi.org/10.1007/978-3-642-34378-0>
- Rao, C. R. (1973). *Linear statistical inference and its applications* (2. Aufl.). Wiley, New York.
- Rauch, G. (2016). Was ist Siri? [Zugegriffen: 06.09.2024]. <https://www.bigdata-insider.de/was-ist-siri-a-572665/>
- Roth, J. (2022). Digitale Lernumgebungen – Konzepte, Forschungsergebnisse und Unterrichtspraxis. In G. Pinkernell, F. Reinhold, F. Schacht & D. Walter (Hrsg.), *Digitales Lehren und Lernen von Mathematik in der Schule: Aktuelle Forschungsbefunde im Überblick* (S. 109–136). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-65281-7_6
- Sanusi, I. T., Oyelere, S. S., Vartiainen, H., Suhonen, J., & Tukiainen, M. (2023). A systematic review of teaching and learning machine learning in K-12 education. *Education and Information Technologies*, 28(5), 5967–5997.
- Sartorius, G. (2018). *Erfassen, Verarbeiten und Zuordnen multivariater Messgrößen: Neue Rahmenbedingungen für das Nächste-Nachbarn-Verfahren* (1. Auflage). Springer Vieweg Wiesbaden. <https://doi.org/10.1007/978-3-658-23576-5>
- Schönbrot, S. (2022). *Optimierungsprobleme in der mathematischen Modellierung : Grundlegende Aspekte und Chancen aus Sicht der Mathematikdidaktik – her- ausgestellt an aktuellen Problemen aus der Forschung zu künstlicher Intelligenz und erneuerbaren Energien* [Diss.]. Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000143530>
- Schönbrot, S., Camminady, T., & Frank, M. (2022). Mathematische Grundlagen der Künstlichen Intelligenz im Schulunterricht: Chancen für eine Bereiche- rung des Unterrichts in linearer Algebra. *Mathematische Semesterberichte*, 69, 73–101. <https://doi.org/10.1007/s00591-021-00310-x>
- Schönbrot, S., Hoeffer, K., & Frank, M. (2024). AI education as a starting point for interdisciplinary STEM projects. *Thirteenth Congress of the European Society for Research in Mathematics Education (CERME13)*.
- Schönbrot, S., Wohak, K., & Frank, M. (2021). Digital tools to enable collaborative mathematical modeling online. *MSEL-Heft*.

- Schukajlow, S., Blum, W., & Krämer, J. (2011). Förderung der Modellierungskompetenz durch selbständiges Arbeiten im Unterricht mit und ohne Lösungsplan. *Praxis der Mathematik in der Schule*, 53(2), 40–45.
- Schüller, K., Busch, P., & Hindinger, C. (2019). Future skills: ein framework für data literacy. *Hochschulforum Digitalisierung*, 46, 1–128.
- Schwarz, B., & Herrmann, P. (2015). Bezüge zwischen Schulmathematik und Linearer Algebra in der hochschulischen Ausbildung angehender Mathematiklehrkräfte – Ergebnisse einer Dokumentenanalyse. *Mathematische Semesterberichte*, 62, 195–217. <https://doi.org/10.1007/s00591-015-0147-7>
- Serfozo, R. (2009). *Basics of applied stochastic processes*. Springer Science & Business Media.
- Shannon, C. E. (1949). Communication in the Presence of Noise. *Proceedings of the IRE*, 37 (1), 10–21.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3), 379–423.
- Shen, H., Jin, H., Cabrera, Á. A., Perer, A., Zhu, H., & Hong, J. I. (2020). Designing alternative representations of confusion matrices to support Non-Expert public understanding of algorithm performance. *Proceedings of the ACM on human-computer interaction*, 4(CSCW2), 1–22. <https://doi.org/10.1145/3415224>
- Siller, H.-S., & Greefrath, G. (2010). Mathematical Modelling in Class regarding to Technology. *Proceedings of the 6th CERME conference*, 2136–2145.
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. <https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002>
- Stark, E., Ueberwasser, S., & Göhring, A. (2020). Korpus "What's up, Switzerland?" [Zugegriffen: 12.07.2021]. www.whatsup-switzerland.ch
- Statistisches Bundesamt. (2024). Anzahl der Studierenden im Fach Informatik in Deutschland nach Geschlecht in den Wintersemestern von 1998/1999 bis 2023/2024. <https://de.statista.com/statistik/daten/studie/732331/umfrage/studierende-im-fach-informatik-in-deutschland-nach-geschlecht/>
- Steffen, N. (2016). *Didaktisch-methodische Ausarbeitung eines Lernmoduls zum Thema Shazam im Rahmen eines mathematischen Modellierungstages für Schülerinnen* [Bachelorarbeit]. RWTH Aachen.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. F. (2020). Learning to summarize from human feedback. *arXiv (Cornell University)*. <http://export.arxiv.org/pdf/2009.01325>
- Tatman, R. (2017). Gender and dialect bias in YouTube's automatic captions. *Proceedings of the first ACL workshop on ethics in natural language processing*, 53–59.
- Tedre, M., Toivonen, T., Vartiainen, H., Jormanainen, I., Valtonen, T., Kahila, J., & Pears, A. (2021). Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. *IEEE Access*, PP, 1–1. <https://doi.org/10.1109/ACCESS.2021.3097962>

- Touretzky, D., & Gardner-McCune, C. (2023). Guiding Students to Investigate What Google Speech Recognition Knows about Language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13), 16040–16047. <https://doi.org/10.1609/aaai.v37i13.26905>
- Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K-12: What Should Every Child Know about AI? *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 9795–9799. <https://doi.org/10.1609/aaai.v33i01.33019795>
- Trabs, M., Jirak, M., Krenz, K., & Reiß, M. (2021). *Statistik und maschinelles Lernen : eine mathematische Einführung in klassische und moderne Methoden*. Springer Spektrum.
- Tulodziecki, G. (2020). Künstliche Intelligenz und Didaktik. *Pädagogische Rundschau*, 74(4), 363–378.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind* 49, 433–460.
- Vartiainen, H., Toivonen, T., Jormanainen, I., Kahila, J., Tedre, M., & Valtonen, T. (2021). Machine learning for middle schoolers: Learning through data-driven design. *International Journal of Child-Computer Interaction*, 29, 100281. <https://doi.org/10.1016/j.ijcci.2021.100281>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv (Cornell University)*. <https://arxiv.org/pdf/1706.03762v5>
- Väyrynen, P. A., Noponen, K., & Seppänen, T. (2007). Analysing performance in a word prediction system with multiple prediction methods. *Computer Speech & Language*, 21(3), 479–491. <https://doi.org/https://doi.org/10.1016/j.csl.2006.09.002>
- Vidal Ruiz, E., Casacuberta Nolla, F., & Rulot Segovia, H. (1985). Is the DTW “distance” really a metric? An algorithm reducing the number of DTW comparisons in isolated word recognition. *Speech Communication*, 4(4), 333–344. [https://doi.org/https://doi.org/10.1016/0167-6393\(85\)90058-5](https://doi.org/https://doi.org/10.1016/0167-6393(85)90058-5)
- Von Hilgers, P., & Langville, A. N. (2006). The five greatest applications of Markov chains. *Proceedings of the Markov Anniversary meeting*, 155–158.
- Vos, P. (2011). What is 'Authentic' in the Teaching of Mathematical Modelling? In G. Kaiser, W. Blum, R. Borromeo Ferri & G. Stillman (Hrsg.), *Trends in Teaching and Learning of Mathematical Modelling - ICTMA14* (S. 713–722).
- Vos, P. (2018). “How Real People Really Need Mathematics in the Real World“ - Authenticity in Mathematics Education. *Education Sciences*, 8. <https://doi.org/10.3390/educsci8040195>
- Wächter, A., & Biegler, L. T. (2006). On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1), 25–57.
- Wandmacher, T. (2008). *Adaptive word prediction and its application in an assistive communication system* [Diss.]. Universität Tübingen. <https://publikationen.uni-tuebingen.de/xmlui/handle/10900/46325>
- Wang, N., & Johnson, M. (2019). AI Education for K-12: Connecting AI Concepts to High School Math Curriculum. *Workshop on Education in Artificial Intel-*

- elligence K-12, 28th International Joint Conference on Artificial Intelligence.* <https://par.nsf.gov/biblio/10440200>
- Warden, P. (2018). Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *ArXiv e-prints*. <https://arxiv.org/abs/1804.03209>
- Wendemuth, A. (2004). *Grundlagen der stochastischen Sprachverarbeitung*. Oldenbourg Wissenschaftsverlag. <https://doi.org/10.1524/9783486595000>
- Winter, H. (1995). Mathematikunterricht und Allgemeinbildung. *Mitteilungen der Gesellschaft für Didaktik der Mathematik*, 61, 37–46. <https://ojs.didaktik-der-mathematik.de/index.php/mgdm/article/view/69/80>
- Wohak, K. (2021). *Didaktisch-methodische Entwicklung und Evaluation von computergestützten Lehr- und Lernmaterialien zur Mathematik inverser Probleme* [Diss.]. Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000136535>
- Yalova, K., Babenko, M., & Yashyna, K. (2023). Automatic Speech Recognition System with Dynamic Time Warping and Mel-Frequency Cepstral Coefficients. *COLINS* (2), 141–151.

A Lernmaterial zum Thema Wortvorhersage

Auf den nachfolgenden Seiten sind alle in dieser Arbeit beschriebenen digitalen Arbeitsblätter mit eingetragenen Lösungen abgedruckt. Alle ausklappbaren Lösungen und Tipps sind hier ausgeklappt. Für eine Exploration der interaktiven Elemente sei auf die Lernplattform¹ verwiesen, auf der die digitalen Arbeitsblätter mit allen zugehörigen Zusatzblättern und Tippkarten in ihrer vollen Funktion eingesehen werden können. Auch für die Einsicht aller weiteren Begleitmaterialien sei an dieser Stelle auf die Lernplattform verwiesen.

Die Notation in den Codefeldern der Arbeitsblätter ist wie folgt zu verstehen:

- Der Platzhalter `#= NaN`; `=#`, der auch in abgewandelter Form erscheinen kann, kennzeichnet die Stellen, an denen die Lernenden Eingaben ergänzen müssen.
- Die Eingaben, die nach diesem Platzhalter folgen, stellen Beispiellösungen dar. Sie bieten Lehrkräften, die das Material im Unterricht oder in Projekten durchführen wollen, eine Kontrollmöglichkeit.

¹siehe workshops.cammp.online

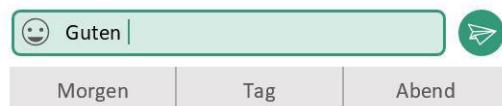
A.1 Arbeitsblatt 1

Arbeitsblatt 1 | Ein erstes Modell



Einleitung

Systeme Künstlicher Intelligenz (KI-Systeme) begegnen uns seit langem als Alexa und Siri, und auch beim täglichen Chatten unterstützt uns ein KI-System: Wir bekommen Wörter vorgeschlagen und sparen damit nicht nur wertvolle Zeit, sondern machen auch weniger Fehler beim Schreiben. Doch woher weiß das Handy, was ich als nächstes schreiben möchte? Wie können solche Wortvorschläge möglichst so generiert werden, dass sie mit hoher Wahrscheinlichkeit das gewünschte Wort liefern? Wie gut funktionieren solche Wortvorschläge?



1) Wortvorhersage an einem kleinen Fallbeispiel

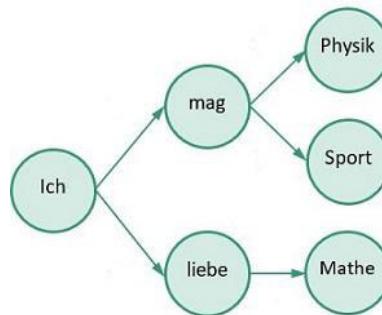
Wir wollen das Generieren von Wortvorschlägen zunächst mithilfe eines kleinen, aus den drei Sätzen

"Ich mag Physik. Ich mag Sport. Ich liebe Mathe."

bestehenden Datensatzes veranschaulichen. Dieser Textausschnitt soll als **Trainingsdatensatz** für ein Modell dienen, das beim Tippen eines Wortes einen Vorschlag für das nächste Wort macht.

Ist das vorherige bereits getippte Wort das Wort "Ich", so sind sinnvolle Vorschläge für das nächste Wort auf Basis dieses Datensatzes beispielsweise die Wörter "mag" oder "liebe", da diese Wörter im Datensatz häufig nach dem Wort "Ich" folgen.

Im Datensatz auftretende Wortfolgen können wir bei einem so kleinen Datensatz noch gut mithilfe eines sogenannten **Übergangsgraphen** darstellen:



Dieser Übergangsgraph ist noch unvollständig. Er zeigt zwar an, von welchem Wort man zu welchem anderen Wort gelangen kann, er zeigt aber noch nicht, mit welcher relativen Häufigkeit auf ein bereits getipptes Wort ein nächstes folgt. Der vollständige Übergangsgraph hilft uns, einen geeigneten Vorschlag für das nächste Wort zu machen, indem wir die relativen Häufigkeiten vergleichen.



a) relative Häufigkeiten bestimmen

Berechne die relativen Häufigkeiten mithilfe des vorgegebenen Textausschnittes: "Ich mag Physik. Ich mag Sport. Ich liebe Mathe.". Notiere die berechneten Häufigkeiten, indem du sie für `NaN` im folgenden Codefeld einsetzt.

 **Schreibweise:** Kommazahlen werden im Code mit einem Punkt geschrieben und Brüche mit einem Schrägstrich.

▼  Tipp 1

Bestimme dazu zunächst die absolute Häufigkeit. Gesucht ist also die Anzahl der Fälle, in denen das Wort „mag“ auf das Wort „Ich“ folgt.

▼  Tipp 2

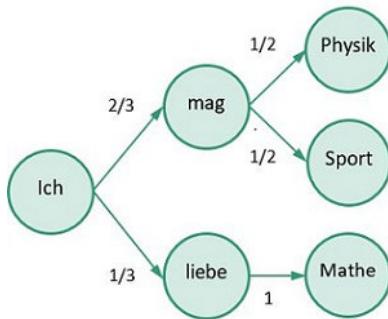
Berechne die relative Häufigkeit. Bestimme dafür zudem die Anzahl der Fälle, in denen das Wort „Ich“ mit einem beliebigen Folgewort auftritt.

```
In [1]: # rel. Häufigkeit von "Ich" zu "mag"
p1= #= NaN; =# 2/3;
# rel. Häufigkeit von "Ich" zu "Liebe"
p2= #= NaN; =# 1/3;
# rel. Häufigkeit von "mag" zu "Physik"
p3= #= NaN; =# 1/2;
# rel. Häufigkeit von "mag" zu "Sport"
p4= #= NaN; =# 1/2;
# rel. Häufigkeit von "Liebe" zu "Mathe"
p5= #= NaN; =# 1;

# Hier nichts ändern!
include("../code/checkAB1.jl"); loaded = false; checkA(p1,p2,p3,p4,p5)
```

✓ Deine Antwort ist korrekt!

Out[1]:



Zur besseren Übersicht können wir die Häufigkeiten auch kompakt in einer Tabelle darstellen. Dies ist vor allem später wichtig, wenn wir größere Trainingsdatensätze betrachten. Die **Übergangstabelle** ist wie folgt aufgebaut:

- Die Zeile steht für das bereits getippte Wort (Wort1), welches auch **Wortvorgeschichte** genannt wird.
- Die Spalte steht für das Folgewort (Wort2).
- Die relativen Häufigkeiten sind bislang noch nicht eingetragen und daher als Fragezeichen dargestellt.

		Wort2					
		Ich	liebe	mag	Mathe	Physik	Sport
Wort1	Ich	?	?	?	?	?	?
	liebe	?	?	?	?	?	?
	mag	?	?	?	?	?	?
	Mathe	?	?	?	?	?	?
	Physik	?	?	?	?	?	?
	Sport	?	?	?	?	?	?

i Die Wörter "Mathe", "Physik" und "Sport" werden, obwohl sie offensichtlich kein Folgewort haben, dennoch in der Übergangstabelle aufgeführt, um die Übergangstabelle in ihrer allgemeinen Form und nicht nur für diesen speziellen Fall gültig darzustellen.



b) Übergangstabelle erstellen

Trage die fehlenden relativen Häufigkeiten im folgenden Codefeld ein, indem du sie für die `NaN` einsetzt. Beachte dabei, dass es Übergänge gibt, die im Trainingsdatensatz nicht auftreten. Die Häufigkeit für einen Übergang ist daher gleich 0.

▼ Tipp

Viele der relativen Häufigkeiten hast du bereits in Aufgabenteil a.) berechnet. Die relative Häufigkeit $\frac{2}{3}$ des Wortes "mag" nach dem Wort "Ich" tragen wir beispielsweise in der Zeile "Ich" und Spalte "mag" ein (s.u.). Übrig bleiben die Übergänge, die im Datensatz nicht auftreten. Was sagt dies über ihre relative Häufigkeit aus?

		Wort2					
		Ich	liebe	mag	Mathe	Physik	Sport
Wort1	Ich	?	?	$\frac{2}{3}$?	?	?
	liebe	?	?	?	?	?	?
	mag	?	?	?	?	?	?
	Mathe	?	?	?	?	?	?
	Physik	?	?	?	?	?	?
	Sport	?	?	?	?	?	?

```
In [2]: # Ersetze NaN
Tabelle = #[NaN NaN NaN NaN NaN NaN; #= [0 1/3 2/3 0 0 0; # erste Zeile der Tabelle
           #= NaN NaN NaN NaN NaN NaN; #= 0 0 0 1 0 0; # zweite Zeile der Tabelle
           #= NaN NaN NaN NaN NaN NaN; #= 0 0 0 0 1/2 1/2; # dritte Zeile der Tabelle
           #= 0 0 0 0 0 0; #= 0 0 0 0 0 0; # vierte Zeile der Tabelle
           #= 0 0 0 0 0 0; #= 0 0 0 0 0 0; # fünfte Zeile der Tabelle
           #= 0 0 0 0 0 0]; #= 0 0 0 0 0 0]; # sechste Zeile der Tabelle

# Hier nichts ändern!
checkB(Tabelle)
```

✓ Deine Antwort ist korrekt!

out[2]:

		Wort2					
		Ich	liebe	mag	Mathe	Physik	Sport
Wort1	Ich	0	$\frac{1}{3}$	$\frac{2}{3}$	0	0	0
	liebe	0	0	0	1	0	0
	mag	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
	Mathe	0	0	0	0	0	0
	Physik	0	0	0	0	0	0
	Sport	0	0	0	0	0	0

2) allgemeine Formel bestimmen

Bei unserem kleinen Trainingsdatensatz ist es noch einfach, die relativen Häufigkeiten per Hand zu ermitteln. Für größere Datensätze ist das jedoch viel zu aufwändig. Entwickle deshalb eine **allgemeine Formel für die relative Häufigkeit**, sodass die Berechnung automatisiert werden kann. Bearbeite dazu folgende Aufgaben:



a) relative Häufigkeit für ein Beispiel

Stelle eine Formel für die relative Häufigkeit auf, mit der das Wort "mag" nach dem Wort "ich" auftritt. Wir wollen diese relative Häufigkeit mit $h_{\text{ich}}(\text{mag})$ bezeichnen.



Schreibweise: Im Workshop wird ab hier mit den klein geschriebenen Wörtern gearbeitet.



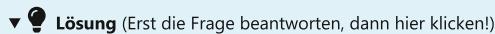
Tipp 1

Die relative Häufigkeit $h_{\text{ich}}(\text{mag})$ kann auch als Häufigkeit aufgefasst werden, mit der die Wortsequenz "ich mag" unter allen Wortsequenzen mit "ich" und einem beliebigen Wort auftritt.



Tipp 2

Bestimme zunächst die absolute Häufigkeit der Wortsequenz "ich mag".



Lösung (Erst die Frage beantworten, dann hier klicken!)

Wir betrachten alle Zweier-Wortsequenzen im Datensatz mit erstem Wort "ich". Um die relative Häufigkeit $h_{\text{ich}}(\text{mag})$ bestimmen zu können, zählen wir, wie häufig die Wortsequenz (ich, mag) auftritt (absolute Häufigkeit). Wir nennen diese Anzahl $N(\text{ich}, \text{mag})$. Für den obigen Datensatz gilt $N(\text{ich}, \text{mag}) = 2$. Außerdem zählen wir, wie häufig das Wort "ich" gefolgt von einem beliebigen Wort auftritt. Wir nennen diese Anzahl $N(\text{ich}, \text{beliebiges Wort})$. Es gelten

$$N(\text{ich}, \text{liebe}) = 1, \\ N(\text{ich}, \text{ich}) = N(\text{ich}, \text{mathe}) = N(\text{ich}, \text{physik}) = N(\text{ich}, \text{sport}) = 0.$$

Die relative Häufigkeit ergibt sich als Quotient aus $N(\text{ich}, \text{mag})$ und $N(\text{ich}, \text{beliebiges Wort})$ zu

$$h_{\text{ich}}(\text{mag}) = \frac{N(\text{ich}, \text{mag})}{N(\text{ich}, \text{ich}) + N(\text{ich}, \text{mathe}) + N(\text{ich}, \text{physik}) + \dots} = \frac{2}{0+2+1+0+0+0} = \frac{2}{3}.$$



b) Wörter zählen

Nenne die Elemente, die im Trainingsdatensatz abgezählt werden müssen, um die relative Häufigkeit $h_{\text{Wort1}}(\text{Wort2})$ berechnen zu können.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Um die relative Häufigkeit $h_{\text{Wort1}}(\text{Wort2})$ bestimmen zu können, ermitteln wir die Anzahl der Wortsequenzen mit "Wort1" und "Wort2": $N(\text{Wort 1, Wort 2})$. Außerdem ermitteln wir die Anzahl der Wortsequenzen, die aus "Wort1" und einem beliebigen Wort aus dem Trainingsdatensatz bestehen: $N(\text{Wort 1, beliebiges Wort})$.



c) Berechnungsformel für die relative Häufigkeit

Stelle eine Formel für die relative Häufigkeit $h_{\text{Wort1}}(\text{Wort 2})$ auf.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Die relative Häufigkeit ergibt sich dann als Quotient aus $N(\text{Wort 1, Wort 2})$ und $N(\text{Wort 1, beliebiges Wort})$ zu

$$h_{\text{Wort1}}(\text{Wort 2}) = \frac{N(\text{Wort 1, Wort 2})}{N(\text{Wort 1, beliebiges Wort})} = \frac{N(\text{Wort 1, Wort 2})}{N(\text{Wort 1, Wort 1}) + N(\text{Wort 1, Wort 2}) + \dots}.$$

3) Die besten Wörter vorschlagen

Wir wollen die relativen Häufigkeiten nun nutzen, um Vorschläge für das nächste Wort zu geben. Nach Tippen des Wortes "ich" wollen wir dasjenige Wort vorschlagen, das am häufigsten beim Chatten nach dem Wort "ich" geschrieben wird. Mit dem Wissen um die relativen Häufigkeiten im Trainingsdatensatz würden wir das Wort "mag" nach dem Wort "ich" mit der höchsten relativen Häufigkeit von $\frac{2}{3}$ vorschlagen.



Modellebene

In unserem Modell fassen wir das Auswählen eines Wortes nach einem bereits geschriebenen Wort als Zufallsexperiment auf. Die Wahl des nächsten Wortes erfolgt entsprechend der relativen Häufigkeit $h_{\text{Wort1}}(\text{Wort 2})$, mit der die Wortsequenz insgesamt in allen Texten, die beim Chatten verfasst wurden, auftritt. Wir nennen diese relative Häufigkeit **Übergangswahrscheinlichkeit**. Diese Übergangswahrscheinlichkeiten kennen wir allerdings nicht. Wir können nur diejenigen relativen Häufigkeiten beobachten, mit der die Wortsequenzen im Trainingsdatensatz auftreten. Wir hoffen, dass die relativen Häufigkeiten, die wir im Trainingsdatensatz beobachten, sich nicht zu stark von den

Übergangswahrscheinlichkeiten unterscheiden. Die **beobachteten relativen Häufigkeiten können wir dann als Näherung für die Übergangswahrscheinlichkeiten verwenden**.

Dafür müssen allerdings geeignete Trainingsdatensätze ausgewählt werden, da die relativen Häufigkeiten stark vom vorliegenden Text abhängen. In unserem Beispiel würde die Übergangswahrscheinlichkeit von "mag" zu "Mathe" zu null genähert werden, obwohl diese Wortsequenz im Sprachgebrauch sicherlich vorkommen kann. Um die Näherung der Übergangswahrscheinlichkeiten durch die relativen Häufigkeiten zu verbessern, müssen umfangreichere Trainingsdatensätze verwendet werden, die den Sprachgebrauch beim Chatten möglichst gut repräsentieren. Wir verwenden daher einen größeren Datensatz von dem wir annehmen, dass er Alltagstexte ausreichend gut repräsentiert.

Dank der aufgestellten Formel können die relativen Häufigkeiten in einem größeren Datensatz automatisiert berechnet und in einer Übergangstabelle, abgespeichert werden. Die Funktion `bestesWort("NaN")` sortiert die Einträge der Zeile des eingegebenen Wortes für `NaN` in der Übergangstabelle. Anschließend gibt die Funktion die drei Wörter mit der höchsten relativen Häufigkeit in absteigender Reihenfolge aus. Wir probieren das zunächst mit dem kleinen Trainingsdatensatz aus.

 Im Code wird mit den kleingeschriebenen Wörtern gearbeitet.



Überprüfe, ob diejenigen Wörter vorgeschlagen werden, die du auf Grundlage des Trainingsdatensatzes ebenfalls vorschlagen würdest, indem du für `NaN` verschiedene Wörter aus dem Trainingsdatensatz einsetzt. Achte auf die Kleinschreibung des Wortes.

Setze anschließend auch ein Wort ein, auf das in unserem Datensatz kein Wort folgt oder ein Wort, das gar nicht im Datensatz auftritt. Beschreibe deine Beobachtung und erkläre, weshalb diese Beobachtung mit der Konstruktion des Modells erklärbar ist.

In [3]: `#= bestesWort_Bi("NaN") =# bestesWort_Bi("ich")`

Out[3]: 2-element Vector{String}:
 "mag"
 "liebe"

4) Mit einem echten Datensatz arbeiten

Wie du eben festgestellt hast, kann das Modell nur für Wörter Vorschläge machen, die auch im Trainingsdatensatz auftreten. Daher ist es wichtig, einen großen Datensatz als Grundlage für unser Modell zu verwenden. Wir nutzen daher einen Datensatz, der aus über 300.000 Wörtern besteht. Dieser wird eingelesen und die relativen Häufigkeiten gemäß der von dir aufgestellten Formel automatisiert ermittelt. Dieser Schritt wird auch als **Training des Modells** bezeichnet.



a) Häufige und seltene Wörter als Wortvorgeschichte

Erprobe auch dieses Modell mit einigen Wörtern, indem du in der Funktion `bestesWort_Bi("NaN")` für `NaN` verschiedene Wörter einsetzt. Beim Ausführen des Codes wird das Vorhersagemodell automatisch erstellt. Beginne zunächst mit Wörtern, die im Sprachgebrauch häufiger vorkommen, und probiere dann auch Wörter aus, die seltener verwendet werden. Beschreibe, was du dabei beobachtest und erläutere, inwiefern es sich dabei um eine Schwäche des Modells handelt.

```
In [4]: # Hier nichts mehr ändern
if loaded == false
    include("../code/checkAB1-Teil3.jl")
    loaded = true
end

#= bestesWort_Bi("NaN") =# bestesWort_Bi("willst")
```

```
Out[4]: 3-element Vector{String}:
"du"
"oder"
"das"
```



b) Wortsequenzen als Wortvorgeschichte

Erprobe das Modell nun für eine Wortvorgeschichte, die länger als ein Wort ist. Setze dazu mehrere Wörter für NaN ein. Prüfe vor allem, ob unterschiedliche Wörter vorgeschlagen werden, wenn das letzte Wort zwar gleich ist, die vorangegangenen Wörter aber unterschiedlich sind (z.B. "weißt du" und "hast du"). Beschreibe deine Beobachtung. Erläutere, weshalb diese Beobachtung mit der Konstruktion des Modells erklärbar ist.

```
In [5]:#= bestesWort_Bi("NaN") =# bestesWort_Bi("weißt du")
```

```
Out[5]: 3-element Vector{String}:
"das"
"hast"
"bist"
```

Schon Fertig?

Zusatz 1)

Auf diesem [Zusatzblatt](#) werden wir die Bildung der Übergangstabellen Schritt für Schritt automatisieren.



Zusatz 2)

Recherchiere, wie viele Wörter der deutsche Wortschatz umfasst. Wie viele verschiedene Zweier-Wortsequenzen, also aus zwei Wörtern bestehende Sequenzen, gibt es somit (ohne die Berücksichtigung grammatischer Regeln)?

Diskussion im Plenum

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!

Anschließend geht es weiter mit [Arbeitsblatt 2](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"¹ und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"².

Autorin: Stephanie Hofmann

¹ Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

² Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.2 Arbeitsblatt 2

Arbeitsblatt 2 | Uni- und Tri-Gramm-Modell



1) Wir erweitern unser Modell: Tri-Gramm-Modell

Auf dem letzten Arbeitsblatt haben wir ein Modell kennengelernt, das nur das vorherige getippte Wort für die Wortvorhersage nutzt. Dafür haben wir die relative Häufigkeit für die Zweier-Wortsequenzen, also Wortsequenzen, die aus dem vorherigen Wort und einem möglichen Folgewort bestehen, betrachtet. Das Modell wird daher als Bi-Gramm-Modell bezeichnet, also als Modell, das auf Zweier-Wortsequenzen (Bi-Grammen) beruht. Wir gehen nun das Problem an, dass unser Modell die vorherigen Wörter nur begrenzt zur Bildung einer Vorhersage nutzt. Dafür erweitern wir unseren Trainingsdatensatz um einen zusätzlichen Satz:

"Ich mag Physik. Ich mag Sport. Ich liebe Mathe. Jeder mag Ferien."

Schön wäre es, wenn nach den Wörtern "Ich mag" die Wörter "Physik" und "Sport" vorgeschlagen werden, nach den Wörtern "Jeder mag" aber das Wort "Ferien" vorgeschlagen wird. Unser Ziel besteht somit darin, ein Modell zu entwickeln, welches in der Lage ist, den erweiterten Kontext eines Satzes für die Wortvorhersage zu nutzen und nicht nur das vorherige Wort zu berücksichtigen.



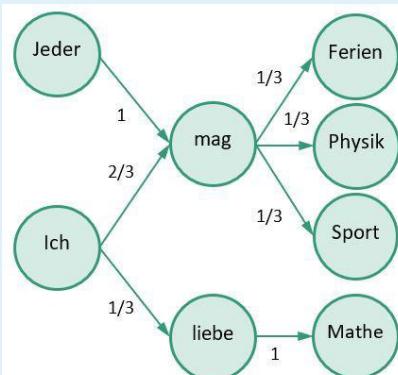
a) Übergangsgraph des Bi-Gramm-Modells an erweiterten Trainingsdatensatz anpassen

Zunächst wollen wir den Übergangsgraphen des Bi-Gramm-Modells an den erweiterten Trainingsdatensatz anpassen. Füge dazu zu deinem bereits existierenden Übergangsgraphen die durch den zusätzlichen Satz neu erzeugten Wege hinzu.



▼ Lösung (Erst die Frage beantworten, dann hier klicken!)

Um den Übergangsgraphen an den erweiterten Trainingsdatensatz anzupassen, werden die beiden Wörter "Jeder" und "Ferien" hinzugefügt sowie die neuen Wege und die relativen Häufigkeiten angepasst:





b) Vorschläge des Bi-Gramm-Modells

Lese am Übergangsgraphen ab, welche Vorschläge unser Bi-Gramm-Modell nach den Wörtern "ich mag" und "jeder mag" gibt.



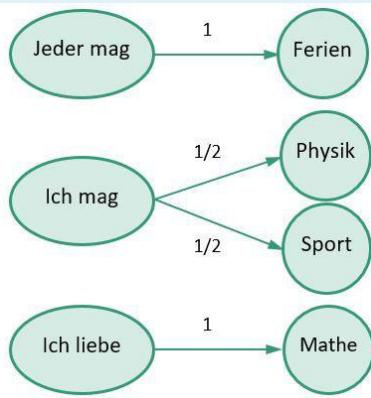
c) Übergangsgraph des Tri-Gramm-Modells

Überlege dir, wie du den Übergangsgraphen verändern musst, wenn die Wortvorgeschichte nicht nur aus einem, sondern aus den beiden vorangegangenen Wörtern besteht. Dieses Modell wird als **Tri-Gramm-Modell** bezeichnet. Zeichne auch diesen Übergangsgraphen auf dein Antwortblatt.



▼ ? Lösung (Erst die Frage beantworten, dann hier klicken!)

Die Berücksichtigung der vorherigen beiden Wörter für eine Wortvorhersage können wir im Übergangsgraphen mit Hilfe von "Päckchen" aus zwei Wörtern realisieren:



Anmerkung: Es ist ebenso korrekt, die Bi-Gramme „mag Ferien“, „mag Physik“, „mag Sport“ beziehungsweise „liebe Mathe“ als Knoten darzustellen, anstatt nur „Ferien“, „Physik“, „Sport“ oder „Mathe“. Dies ist insbesondere dann relevant, wenn der Satz nach diesen Wörtern noch nicht endet und die Bi-Gramme „mag Ferien“, „mag Physik“, „mag Sport“ und „liebe Mathe“ selbst als Wortvorgeschichten für ein weiteres Wort dienen. Da der Satz in diesem Fall jedoch endet, kann auch die oben gewählte Darstellung verwendet werden.



d) Vorschläge des Tri-Gramm-Modells

Lese am Übergangsgraphen ab, welche Vorschläge unser Tri-Gramm-Modell nach den Wörtern "ich mag" und "jeder mag" gibt. Vergleiche die Vorschläge mit denjenigen in Aufgabenteil b).

Zusatz 1) Bildung der Übergangstabelle mit einem Computerprogramm automatisieren

Auf diesem [Zusatzblatt](#) werden wir die Bildung der Übergangstabellen Schritt für Schritt automatisieren.



e) Erproben des Modells

Basierend auf dem großen Trainingsdatensatz wollen wir erproben, welche Vorschläge das Tri-Gramm-Modell ausgibt.

Gib zwei beliebige Wörter für `NaN` ein. Überprüfe dabei, ob unterschiedliche Wörter vorgeschlagen werden, wenn das letzte Wort zwar gleich ist, die vorletzten Wörter aber unterschiedlich sind (z.B. "weißt du" und "hast du").

```
In [1]: # Hier nichts ändern!
include("../code/checkAB2.jl");

# Ersetze NaN

#= bestesWort_Tri("NaN") =# bestesWort_Tri("weißt du")

Out[1]: 3-element Vector{String}:
"was"
"schon"
"wie"
```

2) Wir erweitern unser Modell: Uni-Gramm-Modell

Auf Arbeitsblatt 1 haben wir außerdem festgestellt, dass für Wörter, die im Sprachgebrauch seltener auftreten, häufig keine Vorschläge generiert werden können. Diesem Problem wollen wir uns widmen.

Wenn die bereits geschriebene Wortsequenz nicht im Trainingsdatensatz auftritt, kann mit dem Bi-Gramm- oder Tri-Gramm-Modell kein Vorschlag generiert werden. Eine Möglichkeit, wie trotzdem ein sinnvoller Vorschlag gemacht werden kann, ist die Folgende: Immer dann, wenn die Wortsequenz nicht im Trainingsdatensatz aufgetreten ist, werden die Wörter vorgeschlagen, die im Trainingsdatensatz insgesamt am häufigsten auftreten. Das zugehörige Modell wird als **Uni-Gramm-Modell** bezeichnet.

Zusatz 2) Entwicklung des Uni-Gramm-Modells

Theoretisch weißt du nun, wie ein Uni-Gramm-Modell funktioniert. Auf diesem [Zusatzblatt](#) hast du die Möglichkeit, dir die Entwicklung des Uni-Gramm-Modells mit Hilfe des Computers praktisch zu erarbeiten.



Erproben des Modells

Basierend auf dem großen Trainingsdatensatz wollen wir erproben, welche Vorschläge das Uni-Gramm-Modell ausgibt.

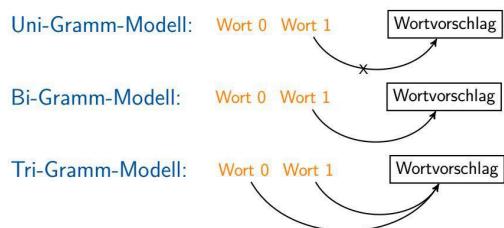
- Lasse dir Wortvorschläge mit Hilfe des Uni-Gramm-Modells ausgeben. Gib dazu ein beliebiges Wort oder mehrere beliebige Wörter für `NaN` ein.
- Was passiert, wenn du die Wortvorgeschichte änderst? Suche nach einer Erklärung für deine Beobachtung.

```
In [2]:#= bestesWort_Uni("NaN") =# bestesWort_Uni("willst")
```

```
Out[2]: 3-element Vector<String>:
"ich"
"und"
"die"
```

3) Vergleich der Modelle

Wir haben drei verschiedene Modelle kennengelernt. Du siehst hier eine Übersicht über alle drei Modelle und deren Funktionsweise:



Mache dir Gedanken über die Vor- und Nachteile eines jeden Modells!



Zusatz 3)

Recherchiere, wie viele Wörter der deutsche Wortschatz umfasst. Wie viele verschiedene Uni-Gramme, wie viele verschiedene Bi-Gramme und wie viele verschiedene Tri-Gramme gibt es somit (ohne die Berücksichtigung grammatischer Regeln)?

Diskussion im Plenum

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!

Anschließend geht es weiter mit [Arbeitsblatt 3](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletistik" des Korpus "LIMAS"2.

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung).
Universität Bonn, Universität Regensburg.

A.3 Arbeitsblatt 3

Arbeitsblatt 3 | Kombination der Modelle



Wir haben festgestellt, dass **jedes Modell unterschiedliche Vorteile** hat und die Performance der Modelle stark von der jeweiligen Wortvorgeschichte abhängt. Um ein Vorhersagemodell zu konstruieren, das die Vorteile aller N-Gramm-Modelle bündelt, ist es daher nötig, die **Modelle zu kombinieren**.

1) Verknüpfung der N-Gramm-Modelle über einen gewichteten Mittelwert

Um bei jeder Wortvorgeschichte den Einfluss aller Modelle zu berücksichtigen, verwenden wir als Übergangswahrscheinlichkeit einen Mittelwert der Übergangswahrscheinlichkeiten im Uni-, Bi- und Tri-Gramm-Modell. Wir nennen sie kurz Uni-, Bi- und der Tri-Gramm-Wahrscheinlichkeit. Dabei können die drei Modelle mit einem unterschiedlichen Gewicht im Mittel berücksichtigt werden. Einen solchen gewichteten Mittelwert berechnest du beispielsweise, wenn du deine Gesamtnote in einem Kurs ermitteln möchtest, in dem verschiedene Leistungen unterschiedlich gewichtet werden. Eine Klassenarbeit zählt zum Beispiel mehr als ein Test.

! Vorsicht: Die Notenbildung ist natürlich keine rein mathematische Berechnung, sondern eine pädagogisch fachliche Gesamtwertung.



a) gewichteter Mittelwert am Beispiel einer Durchschnittsnote

Berechne die Gesamtnote eines Kurses eines Schülers/ einer Schülerin, wenn jede Klassenarbeit ein Viertel der Gesamtnote, die mündliche Note 0,4 der Gesamtnote und die Tests 0,1 der Gesamtnote ausmachen. Folgende Noten wurden geschrieben:

Leistung	Note
Klassenarbeit 1	3,0
Klassenarbeit 2	2,0
mündliche Note	2,0
Test	1,0

🔧 Schreibweise: Du kannst das Codefeld als Taschenrechner nutzen. Kommazahlen werden im Code mit einem Punkt geschrieben, Brüche mit einem Schrägstrich `/` und ein Malzeichen wird durch ein Sternchen `*` dargestellt.

▼ Tipp

Ein einfaches Addieren der Noten führt hier zu einem falschen Ergebnis, da die einzelnen Noten mit einem unterschiedlichen Gewicht in die Gesamtnote eingehen.

▼ Tipp

Die Noten müssen mit dem entsprechenden Gewichtungsfaktor multipliziert werden, damit sie unterschiedlich

stark zur Gesamtnote beitragen.

```
In [1]: # Ersetze NaN
gesamtnote = #= NaN; =# 1/4 * 3.0 + 1/4 * 2.0 + 0.4 * 2.0 + 0.1 * 1.0

# Hier nichts ändern!
include("../code/checkAB3.jl")
checkC(gesamtnote)

✓ Deine Antwort ist korrekt!
```

Out[1]: 2.15



b) allgemeine Formel für den gewichteten Mittelwert

Wir möchten nun eine allgemeine Formel für den gewichteten Mittelwert aufstellen, um den Sachverhalt nachher auf die Übergangswahrscheinlichkeiten übertragen zu können.

Formuliere eine allgemeine Formel für den gewichteten Mittelwert mit vorgegebenen Gewichten g_1, g_2, g_3 (bei der Gesamtnote der jeweilige Anteil einer Note an der Gesamtnote) und vorgegebenen Werten n_1, n_2, n_3 (bei der Gesamtnote die einzelnen Noten). Welche Eigenschaft erfüllt die Summe aller Gewichte?

▼💡 Lösung (Erst die Frage beantworten, dann hier klicken!)

gewichteter Mittelwert:

$$M = g_1 \cdot n_1 + g_2 \cdot n_2 + g_3 \cdot n_3$$

Alle Gewichte müssen zusammen ein Ganzes ergeben. Die Summe der Gewichte g_1, g_2 und g_3 ist daher eins. Als Anteile eines Ganzen gilt für die Gewichte zudem, dass sie Werte von null bis eins annehmen können.



c) gewichteter Mittelwert der Übergangswahrscheinlichkeiten

Ganz genau so gehen wir jetzt vor, um den gewichteten Mittelwert der Übergangswahrscheinlichkeit für ein Wort zu berechnen. Wir nennen diesen **Gesamtübergangswahrscheinlichkeit**. Wir addieren die einzelnen Übergangswahrscheinlichkeiten des Uni-, Bi- und Tri-Gramm Modells und gewichten sie entsprechend ihrer Bedeutung für die Gesamtübergangswahrscheinlichkeit.

Stelle eine Formel für die Gesamtübergangswahrscheinlichkeit auf. Verwende dazu die Wahrscheinlichkeiten in den einzelnen Modellen $P_{Uni}(Wort)$, $P_{Bi}(Wort)$, $P_{Tri}(Wort)$ und die Gewichte g_1, g_2, g_3 .

- i** Die Bi- und Tri-Gramm-Wahrscheinlichkeiten hängen von dem einen vorangegangenen Wort bzw. von den beiden vorangegangenen Wörtern ab. Um eine einfachere Darstellung zu gewährleisten, verzichten wir bei der Schreibweise $P_{Bi}(Wort)$ und $P_{Tri}(Wort)$ jedoch darauf, die Abhängigkeit von der Wortvorgeschichte darzustellen.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Die Gesamtübergangswahrscheinlichkeit ergibt sich zu

$$P_{ges}(Wort) = g_1 \cdot P_{Uni}(Wort) + g_2 \cdot P_{Bi}(Wort) + g_3 \cdot P_{Tri}(Wort).$$



d) Wahl der Gewichte

Stelle erste Überlegungen an, wie du die Gewichte für die einzelnen Übergangswahrscheinlichkeiten wählen würdest. Begründe deine Überlegungen. Kannst du bereits mögliche Werte für die Gewichte ausschließen?

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

- Die Gewichte sollen einen Anteil eines Ganzen darstellen und sind daher zwischen null und eins. Die Werte null und eins schließen wir explizit aus, da andernfalls eines der N-Gramm-Modelle gar nicht berücksichtigt werden würde.
- Die Summe der Gewichte g_1, g_2 und g_3 ist eins.
- Alle weiteren Überlegungen besprechen wir im Plenum.

Wir haben uns Gedanken über die Formel der Gesamtübergangswahrscheinlichkeit und die Wahl der Gewichte gemacht. In der nächsten Aufgabe wollen wir unsere Überlegungen umsetzen und die Gesamtübergangswahrscheinlichkeit konkret für eine Wortvorgeschichte und ein Folgewort ermitteln. Die einzelnen Übergangswahrscheinlichkeiten des Uni-, Bi- und Tri-Gramm-Modells werden dabei wieder durch die relativen Häufigkeiten des Trainingstextes genähert.



e) Gesamtübergangswahrscheinlichkeit an einem Beispiel

Bilde die Gesamtübergangswahrscheinlichkeit der Wortvorgeschichte "ich würde gerne" und des Folgewortes "einfach". Verwende Werte für die Gewichte gemäß der obigen Überlegungen. Verwende die Uni-, Bi- und Tri-Grammwahrscheinlichkeiten.

 Die Uni-, Bi- und Tri-Grammwahrscheinlichkeiten werden im Code über die Funktionen `P_Uni(f)`, `P_Bi(w_1,f)`, `P_Tri(w_2,w_1,f)` aufgerufen. Dabei stehen `f` für das Folgewort, `w_1` für das letzte Wort der Wortvorgeschichte und `w_2` für das vorletzte Wort der Wortvorgeschichte.

 Tipp

Das Wort "ich" muss weder beim Tri-Gramm-Modell noch bei den anderen beiden N-Gramm-Modellen berücksichtigt werden.

```
In [2]: # Gewichte festlegen
g_1 = #= NaN; =# 1/6
g_2 = #= NaN; =# 2/6
g_3 = #= NaN; =# 1-g_1-g_2

# Gesamtübergangswahrscheinlichkeit bestimmen
P_ges = #= NaN; =# g_1*P_Uni("einfach") + g_2*P_Bi("gerne", "einfach") + g_3*P_Tri("würde", "gerne", "einfach")

# Hier nichts ändern!
checkD(g_1,g_2,g_3,P_ges)

✓ Deine Antwort ist korrekt!
```

```
Out[2]: 0.2553
```

Eine solche Gesamtübergangswahrscheinlichkeit muss für jedes mögliche Folgewort ermittelt werden, um entscheiden zu können, welches Folgewort unter dem kombinierten Modell die höchste Gesamtübergangswahrscheinlichkeit hat. Drei der Folgewörter, die im Modell die höchste Gesamtübergangswahrscheinlichkeit haben, können anschließend wie gewohnt vorgeschlagen werden.



f) Erproben des kombinierten Modells

Lasse dir die Wortvorschläge zur Wortvorgeschichte "ich würde gerne" ausgeben. Teste unterschiedliche Gewichte und überprüfe, welche Wahl der Gewichte die für dich sinnvollsten Vorschläge gibt.

 Verwende die Funktion `bestesWort_MW(w,g_1,g_2,g_3)`. Sie berechnet nach obigem Schema die Gesamtübergangswahrscheinlichkeit für jedes mögliche Folgewort einer Wortvorgeschichte `w` und gibt die drei Folgewörter mit der größten Gesamtübergangswahrscheinlichkeit aus. Dabei steht

- `w` für die Wortvorgeschichte und wird mit Anführungszeichen "" eingegeben (z.B. "magst du"),
- `g_1`, `g_2`, `g_3` für die Gewichte (achte darauf, Kommazahlen im Code mit Punkt und Brüche mit Schrägstrich / einzugeben).

```
In [3]: # Ersetze NaN
#= bestesWort_MW("NaN",NaN,NaN,NaN) =# bestesWort_MW("ich würde gerne",1/6,2/6,3/6)
```

```
Out[3]: 3-element Vector{String}:
"für"
"einfach"
"in"
```

Schon Fertig?

Auf diesem [Zusatzblatt](#) hast du die Möglichkeit, eine weitere Kombinationsmöglichkeit kennenzulernen: die Rückfallstrategie.

Diskussion im Plenum

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!

Anschließend geht es weiter mit [Arbeitsblatt 4](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"2.

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.4 Arbeitsblatt 4

Arbeitsblatt 4 | Bewertung



Auf diesem Arbeitsblatt beschäftigen wir uns mit der **Bewertung von Sprachmodellen**. Wir haben bereits die Kreuzentropie als Qualitätsmaß kennengelernt. Nun wollen wir diese für das kombinierte N -Gramm-Modell mit gewichtetem Mittelwert berechnen.

Erinnerung: Kreuzentropie

Die **Kreuzentropie** ist definiert als

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \dots, \text{Wort } n))$$

mit der **Modellwahrscheinlichkeit** $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \dots, \text{Wort } n)$ (Wahrscheinlichkeit eines Testdatensatzes Wort 1, Wort 2, ..., Wort n unter einem vorgegebenen Modell) und der Anzahl der Wörter im Testdatensatz n .

Eine kleine Kreuzentropie bedeutet, dass das Sprachmodell dem Testdatensatz eine hohe Wahrscheinlichkeit zuweist. Das Sprachmodell passt dann gut zum Testdatensatz und modelliert den Sprachgebrauch für diese Art von Texten gut.

 Falls du die Logarithmusfunktion noch nicht kennst, kannst du dir [hier](#) die Information zur Logarithmusfunktion ansehen.



1) Bewertung unseres Sprachmodells

Wir wollen nun unser letztes Modell, das kombinierte N -Gramm-Modell mit gewichtetem Mittelwert, mit Hilfe der **Kreuzentropie** bewerten. Dazu berechnen wir zunächst die Modellwahrscheinlichkeit.

Erinnerung: Modellwahrscheinlichkeit

Die **Modellwahrscheinlichkeit** gibt an, wie wahrscheinlich ein sogenannter **Testdatensatz** aus dem zugrunde liegenden Sprachmodell entstehen kann. Dabei können wir das Erzeugen des Testdatensatzes aus unserem Sprachmodell als mehrstufiges Zufallsexperiment auffassen. Nach jedem Wort hat das darauffolgende Wort eine gewisse Wahrscheinlichkeit, mit der es auftritt. Diese Wahrscheinlichkeit entspricht der Übergangswahrscheinlichkeit in unserem Modell. Für jedes Wort im Text wird die Übergangswahrscheinlichkeit auf der Grundlage der vorangegangenen Wörter berechnet.

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?



a) Die Modellwahrscheinlichkeit

Stelle die Modellwahrscheinlichkeit $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \dots, \text{Wort } n)$ als Ausdruck aus den Übergangswahrscheinlichkeiten $P_{ges}(\text{Wort 1}), P_{ges}(\text{Wort 2}), \dots, P_{ges}(\text{Wort } n)$ jedes einzelnen Wortes im Text dar.

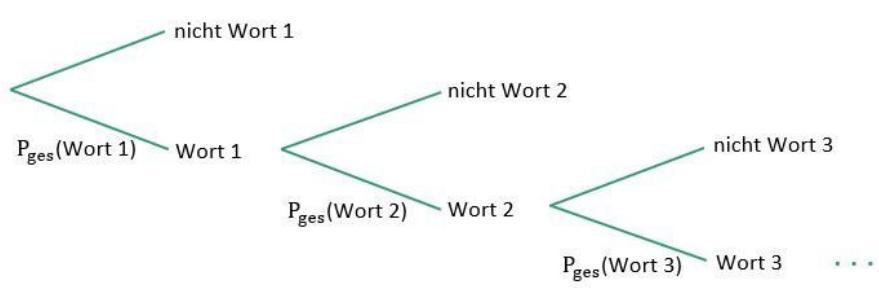
i Die Gesamtwahrscheinlichkeiten hängen von den beiden vorangegangenen Wörtern ab. Um eine einfachere Darstellung zu gewährleisten, verzichten wir bei der Schreibweise $P_{ges}(\text{Wort})$ darauf, die Abhängigkeit von der Wortvorgeschichte darzustellen.

▼ Tipp 1

Das Auftreten jedes Wortes im Text kann als ein einzelnes Ereignis angesehen und die Übergangswahrscheinlichkeit als eine Wahrscheinlichkeit im Baumdiagramm dargestellt werden. Erinnerst du dich noch an die Pfadregeln bei mehrstufigen Zufallsexperimenten?

▼ Tipp 2

Lässt man die Pfade, die nicht interessieren weg, so sieht das Baumdiagramm wie unten aus. Wende nun die 1. Pfadregel (auch Produktregel genannt) an.



▼ Lösung (Erst die Frage beantworten, dann hier klicken!)

Das Auftreten jedes Wortes im Text kann als ein einzelnes Ereignis angesehen und die Übergangswahrscheinlichkeit als eine Wahrscheinlichkeit im Baumdiagramm dargestellt werden. Die

Gesamtwahrscheinlichkeit ergibt sich gemäß der 1. Pfadregel oder auch Produktregel für mehrstufige Zufallsexperimente aus dem Produkt der einzelnen Übergangswahrscheinlichkeiten:

$$\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \dots, \text{Wort } n) = P_{ges}(\text{Wort 1}) \cdot P_{ges}(\text{Wort 2}) \cdot \dots \cdot P_{ges}(\text{Wort } n)$$

Die Kreuzentropie ergibt sich somit zu

$$K = -\frac{1}{N} \cdot \log(P_{ges}(\text{Wort 1}) \cdot P_{ges}(\text{Wort 2}) \cdot \dots \cdot P_{ges}(\text{Wort } n)).$$

Bevor wir die Kreuzentropie berechnen, wollen wir uns zunächst anschauen, weshalb hierfür die Modellwahrscheinlichkeit logarithmiert wird. Dazu nehmen wir die Modellwahrscheinlichkeit genauer unter die Lupe.

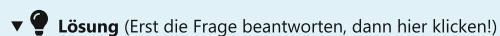


b) Größe der Modellwahrscheinlichkeit

Begründe, weshalb die Modellwahrscheinlichkeit für ein Testdatensatz mit mehreren Wörtern immer sehr klein ist.



Welche Werte kann eine Übergangswahrscheinlichkeit annehmen, und welche Konsequenzen hat das für ein Produkt mehrerer Übergangswahrscheinlichkeiten?



Um die Modellwahrscheinlichkeit zu berechnen, werden die einzelnen Übergangswahrscheinlichkeiten, die zwischen null und eins liegen, miteinander multipliziert. Jedes Mal wenn mit einem Faktor kleiner 1 multipliziert wird, wird das Ergebnis kleiner. Aufgrund der großen Anzahl und der somit sehr vielen Multiplikationen ist die Modellwahrscheinlichkeit am Ende sehr klein.

Die Zahl kann sogar so klein werden, dass der Computer die Zahl aufgrund der vielen Stellen nach dem Komma gar nicht mehr genau darstellen kann und sie auf null abrundet. Deshalb Logarithmieren wir die Modellwahrscheinlichkeit. Durch das Logarithmieren der Modellwahrscheinlichkeit kann das Produkt der einzelnen Übergangswahrscheinlichkeiten als Summe dargestellt werden. Eine Summe mit kleinen Zahlen kann vom Computer mit weniger Rundungsfehlern berechnet werden.

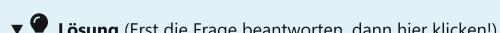


c) Logarithmus der Modellwahrscheinlichkeit

Forme den Ausdruck

$\log(P_{ges}(\text{Wort 1}) \cdot P_{ges}(\text{Wort 2}) \cdot \dots \cdot P_{ges}(\text{Wort } n))$

mithilfe der Logarithmusgesetze zu einer Summe um.



Wir nutzen die Produktregel des Logarithmus, um den Ausdruck umzuformen:

$$\begin{aligned} & \log(P_{ges}(\text{Wort 1}) \cdot P_{ges}(\text{Wort 2}) \cdots P_{ges}(\text{Wort } n)) \\ &= \log(P_{ges}(\text{Wort 1})) + \log(P_{ges}(\text{Wort 2})) + \dots + \log(P_{ges}(\text{Wort } n)) \end{aligned}$$

Für das Weiterrechnen bietet es sich an, mit Summen statt mit Produkten zu arbeiten, da diese vom Computer einfacher und mit weniger Rundungsfehlern berechnet werden können. Durch das Logarithmieren der Modellwahrscheinlichkeit konnten wir dies gewährleisten.

Die Kreuzentropie ergibt sich somit zu

$$K = -\frac{1}{N} \cdot [\log(P_{ges}(\text{Wort 1})) + \log(P_{ges}(\text{Wort 2})) + \dots + \log(P_{ges}(\text{Wort } n))].$$

Wir werden nun Schritt für Schritt erarbeiten, wie logarithmierte Modellwahrscheinlichkeit ermittelt werden kann. Dazu erkunden wir zunächst unseren in einer Liste abgespeicherten Testdatensatz `data_test`. Diese Liste enthält alle Wörter jedes Satzes als ein Element.



d) Den Testdatensatz erkunden

Führe das Codefeld aus, um dir die ersten 3 Elemente der Liste ausgeben zu lassen und die Struktur der Liste zu erkunden.

```
In [1]: #Hier nichts ändern!
include("../code/checkAB4.jl");

data_test[1:3]
```

```
Out[1]: 3-element Vector{Any}:
["den", "schaffner", "war", "jung", "und", "er", "kniff", "ein", "auge", "zu", "als", "ich", "ihm",
"meine", "schülerlegitimation", "zeigte"]
["lothan", "erwidert", "nichts", "auf", "diese", "worte"]
["zu", "erklären", "versuchen", "müssen", "dann", "erst", "würde", "er", "ihn", "bitten", "ein", "paa
r", "tage", "dableiben", "zu", "dürfen"]
```



e) Auf Elemente im Testdatensatz zugreifen

Lasse dir nun das dritte Wort im zweiten Satz ausgeben.

Schreibweise: Du kannst auf die Listeneinträge mit dem Listennamen und einer eckigen Klammer mit der Satz-, bzw. Wortposition in der Liste zugreifen: `data_test[Satzposition][Wortposition]`.

```
In [2]: # Ersetze NaN
w_2_3 = #= NaN; =# data_test[2][3]

# Hier nichts ändern!
checkA(w_2_3)

✓ Deine Antwort ist korrekt!
```

```
Out[2]: "nichts"
```



f) Übergangswahrscheinlichkeit eines konkreten Wortes im Testdatensatz

Ermittle die Gesamtübergangswahrscheinlichkeit des dritten Wortes im zweiten Satz. Verwende hierzu den gewichteten Mittelwert mit den Gewichten $g_1 = \frac{1}{6}, g_2 = \frac{2}{6}, g_3 = \frac{3}{6}$.

 **Variablen und Schreibweise:**

- Auf die N -Grammwahrscheinlichkeiten kannst du im Code mit `P_Uni(f)`, `P_Bi(w_1,f)` und `P_Tri(w_2,w_1,f)` zugreifen.
- Auf den Testdatensatz kannst du im Code mit `data_test` zugreifen.
- Brüche werden im Code mit einem Schrägstrich `/` geschrieben und ein Malzeichen wird durch ein Sternchen `*` dargestellt.

In [3]: `# Ersetze NaN`

```
P_2_3 = ## NaN; =# 1/6*P_Uni(data_test[2][3])+2/6*P_Bi(data_test[2][2],data_test[2][3])+3/6*P_Tri(data_
# Hier nichts ändern!
checkB(P_2_3)
```

 Deine Antwort ist korrekt!

Out[3]: `0.6113`



g) Übergangswahrscheinlichkeit eines beliebigen Wortes im Testdatensatz

Ermittle die Gesamtübergangswahrscheinlichkeit des j -ten Wortes im i -ten Satz. Ersetze dazu die `NaN` im untenstehenden Codefeld. Verwende auch hier den gewichteten Mittelwert mit den Gewichten $g_1 = \frac{1}{6}, g_2 = \frac{2}{6}, g_3 = \frac{3}{6}$. (Wir setzen zunächst voraus, dass das Wort eine Wortvorgeschichte besitzt und nicht unmittelbar am Satzanfang steht.)

 **Variablen und Schreibweise:**

- Auf die N -Grammwahrscheinlichkeiten kannst du im Code mit `P_Uni(f)`, `P_Bi(w_1,f)` und `P_Tri(w_2,w_1,f)` zugreifen.
- Auf den Testdatensatz kannst du im Code mit `data_test` zugreifen.
- Die Gewichte können im Code als Zahl oder Variable `g_1`, `g_2`, `g_3` eingesetzt werden.
- Brüche werden im Code mit einem Schrägstrich `/` geschrieben und ein Malzeichen wird durch ein Sternchen `*` dargestellt.

In [4]: `# Ersetze NaN`

```
P_i_j = ## NaN; =# 1/6* P_Uni(data_test[i][j]) + 2/6 * P_Bi(data_test[i][j-1],data_test[i][j]) + 3/6 *
# Hier nichts ändern!
checkC(P_i_j)
```

 Deine Antwort ist korrekt!



h) Berechnung der Modellwahrscheinlichkeit

Berechne nun die logarithmierte Modellwahrscheinlichkeit des Testdatesatzes. Ersetze dazu die `NaN` im untenstehenden Codefeld.

 **Variablen und Schreibweise:**

- Das Folgewort und die Wörter der Wortvorgeschichte müssen dabei nicht mehr aus dem Datensatz ermittelt werden, sondern können direkt über `f`, `w_1` bzw. `w_2` im Code eingegeben werden.
- Der Logarithmus der Modellwahrscheinlichkeit wird im Code in der Variable `P_Mod` gespeichert.
- Der Logarithmus einer Zahl x kann im Code mit `log(x)` eingegeben werden.
- Zählungen beginnen bei `1`.
- Mit `length(data_test)` kannst du auf die Länge des gesamten Testdatensatzes, mit `length(data_test[i])` auf die Länge des i -ten Eintrages (des i -ten Satzes) zugreifen.

 **Tipp zum Start- und Endwert der for-Schleife**

▼  Tipp zur Berechnung der Übergangswahrscheinlichkeit

Achte darauf, dass das Folgewort und die vorangegangenen Wörter bereits im Code als `f`, `w_1`, `w_2` definiert sind.

▼  Tipp zur Berechnung der logarithmierten Modellwahrscheinlichkeit

Achte darauf, dass du die logarithmierten Übergangswahrscheinlichkeiten zu `P_Mod` hinzufügst und `P_Mod` nicht überschreibst. Schaue dir dazu das Beispiel [hier](#) noch einmal an.

```
In [5]: # Hier nichts ändern!
include("../code/changeAB4.jl");

# Ersetze die NaN
P_Mod = 0 # Variable in der die logarithmierte Modellwahrscheinlichkeit gespeichert wird

for i in #= NaN : NaN =# 1 : length(data_test) # alle Sätze im Testdatensatz data_test durchlaufen: Gib den

    for j in #= NaN : NaN =# 1 : length(data_test[i]) # alle Wörter eines Satzes durchlaufen: Gib den

        # Folgewort f und vorherige Wörter w_1 und w_2 bestimmen
        f = data_test[i][j]
        if j-1>0
            w_1 = data_test[i][j-1]
        else
            w_1 = ""
        end
        if j-2>0
            w_2 = data_test[i][j-2]
        else
            w_2 = ""
        end

        # Übergangswahrscheinlichkeit der Wörter berechnen
        P_i_j = #= NaN =# 1/6 *P_Uni(f) + 2/6 *P_Bi(w_1,f) + 3/6 *P_Tri(w_2,w_1,f)

        # Berechnung der Logarithmierten Modellwahrscheinlichkeit

        P_Mod = #= NaN =# P_Mod + log(P_i_j)

    end
end
```

```
# Hier nichts ändern!
checkD(P_Mod)

✓ Deine Antwort ist korrekt!
Out[5]: -Inf
```



i) Interpretation der logarithmierten Modellwahrscheinlichkeit

Dieses Ergebnis ist dieses Mal nicht das Resultat von Rundungsfehlern. Begründe, weshalb die logarithmierte Modellwahrscheinlichkeit $-\infty$ ist.

▼ ⓘ Tipp 1

Sieh dir den Graphen der Logarithmusfunktion an. Was passiert, wenn die x -Werte immer kleiner werden und sich dem Wert null annähern? [Hier](#) findest du den Graphen der Logarithmusfunktion.

▼ ⓘ Tipp 2

Am Graphen erkennen wir, dass die Funktionswerte immer weiter ins Negative gehen, je kleiner x wird. Für $x = 0$ gibt es genau genommen keinen zugehörigen Funktionswert. Die Funktion ist hier nicht definiert. Der Computer verwendet für die Berechnung des Funktionswertes an der Stelle null deshalb sehr kleine Werte, die nahe bei null liegen. Da der Logarithmus für sehr kleine x immer negativere Werte annimmt, gibt der Computer als Annäherung $-\infty$ aus. Die zugehörige Modellwahrscheinlichkeit, die hier vorliegt, ist somit null.

Was könnte der Grund dafür sein, dass die Modellwahrscheinlichkeit null ist?

▼ ⓘ Tipp 3

Welche Wahrscheinlichkeit wird einem Wort zugewiesen, das nicht im Trainingsdatensatz aufgetreten ist? Wie wirkt sich das auf die Modellwahrscheinlichkeit aus?

▼ ⓘ Lösung (Erst die Frage beantworten, dann hier klicken!)

Du hast die obige Aufgabe richtig gelöst und erhältst dennoch $-\infty$ als Lösung für die logarithmierte Modellwahrscheinlichkeit. Das hat den folgenden einfachen Grund: Im Moment werden Wörter, die nicht im Trainingsdatensatz vorkommen, mit der Auftretenswahrscheinlichkeit null versehen. Diese Wörter würden somit nie von unserem Modell vorgeschlagen werden, und der Testtext würde damit nie aus unserem Modell entstehen. Die logarithmierte Modellwahrscheinlichkeit ergibt sich somit zu $\log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \dots, \text{Wort } n)) = \log(0)$. Der Computer kann den Logarithmus von null nicht direkt berechnen, weil die Logarithmusfunktion an der Stelle null nicht definiert ist. Stattdessen nähert er sich mit sehr kleinen positiven Zahlen dem Wert null, und da der Logarithmus für solche Werte immer stärker ins Negative geht, zeigt der Computer $-\infty$ an.

Jedes Wort tritt aber mit einer gewissen - wenn vielleicht auch kleinen - Wahrscheinlichkeit auf. Um dies zu realisieren wird jedem Wort eine kleine Wahrscheinlichkeit zugeordnet. Man bezeichnet diesen Vorgang als **Glätten**.

Dem Glätten liegt die Idee zugrunde, dass beobachtete Wörter etwas von ihrer Wahrscheinlichkeitsmasse an ungesehene Wörter abtreten, damit deren Wahrscheinlichkeit nicht mehr null ist. Man kann sich das wie beim Kuchenbacken vorstellen, wenn am Ende die Creme auf dem Kuchenboden verteilt wird. Damit alle Stellen des Kuchenbodens mit Creme bedeckt sind, müssen wir die Creme glattstreichen. Wir nehmen dazu an anderen Stellen ein klein wenig Creme weg, um sie zu den Stellen zu schieben, die noch nicht bedeckt sind. Bei den Sprachmodellen lösen wir das durch das Hinzufügen einer kleinen Wahrscheinlichkeit, die für jedes Wort im

Vokabular gleich groß ist. Damit die Summe der Gewichte trotzdem eins bleibt, muss natürlich etwas von den Gewichten der N -Gramm-Modelle an das Gewicht g_0 dieser Glättungswahrscheinlichkeit P_G abgegeben werden. Wir bezeichnen diesen Anteil mit ϵ . Der gewichtete Mittelwert der Wahrscheinlichkeit ergibt sich damit zu

$$P_{ges}(Wort) = g_0 \cdot P_G + (g_1 - \epsilon) \cdot P_{Uni}(Wort) + (g_2 - \epsilon) \cdot P_{Bi}(Wort) + (g_3 - \epsilon) \cdot P_{Tri}(Wort).$$



j) Bildung der Glättungswahrscheinlichkeit

Wir bezeichnen die Anzahl der Wörter im Vokabular mit der Vokabulargröße V . Wie ergibt sich die Glättungswahrscheinlichkeit aus der Vokabulargröße V , wenn allen Wörtern im Vokabular die gleiche Glättungswahrscheinlichkeit zugeordnet werden soll? Stelle eine Formel für die Glättungswahrscheinlichkeit P_G auf.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Da jedem Wort die gleiche Wahrscheinlichkeit zugeordnet werden soll, liegt eine Gleichverteilung vor. Es gibt V Wörter, auf die wir die Wahrscheinlichkeit von eins gleichmäßig aufteilen müssen. Die Glättungswahrscheinlichkeit ergibt sich somit zu

$$P_G = \frac{1}{V}$$



k) Berechnung der Modellwahrscheinlichkeit mit Glättung

Berechne jetzt die logarithmierte Modellwahrscheinlichkeit des Testdatensatzes mit den umverteilten Gewichten $g_{neu0} = 3/18, g_{neu1} = 2/18, g_{neu2} = 5/18, g_{neu3} = 8/18$. Ersetze dazu die `NaN` im untenstehenden Codefeld.

🔧 **Variablen:**

- Die Gewichte können im Code als Zahl oder Variable `g_neu0`, `g_neu1`, `g_neu2`, `g_neu3` eingesetzt werden.
- Wir nehmen an, dass die Vokabulargröße V der Zahl 75000 entspricht, welche als Größe des deutschen Standardwortschatzes gilt.

```
In [6]: # Hier nichts ändern!
include("../code/load_test.jl"); include("../code/checkAB4.jl");

# Ersetze die NaN

P_Mod = 0 # Variable in der die Logarithmierte Modellwahrscheinlichkeit gespeichert wird

for i in 1 : length(data_test) # alle Sätze im Testdatensatz data_test durchlaufen: Gib den Startwert an

    for j in 1 : length(data_test[i]) # alle Wörter eines Satzes durchlaufen: Gib den Startwert und den Endwert an

        # Folgewort f und vorherige Wörter w_1 und w_2 bestimmen
        f = data_test[i][j];
        if j>0
            w_1 = data_test[i][j-1]
        else
            w_1 = ""
        end
    end
end
```

```
if j-2>0
    w_2 = data_test[i][j-2]
else
    w_2 = ""
end

# Übergangswahrscheinlichkeit der Wörter berechnen

P_i_j = ## NaN =# 3/18* 1/75000+ 2/18 *P_Uni(f) + 5/18 *P_Bi(w_1,f) + 8/18 *P_Tri(w_2,w_1,f)

# Berechnung der Logarithmierten Modellwahrscheinlichkeit

P_Mod = P_Mod + log(P_i_j)

end
end
checkE(P_Mod)

✓ Deine Antwort ist korrekt!
-124306.1464
Dies entspricht einer Modellwahrscheinlichkeit nahe 0.0.
```



I) Berechnung der Kreuzentropie

Berechne die Kreuzentropie mit Hilfe der oben ermittelten Größen `P_Mod` (logarithmierte Modellwahrscheinlichkeit) und `n` (Anzahl der Wörter im Testdatensatz).

```
In [7]: # Ersetze NaN

K = ## NaN =# -1/n*P_Mod

# Hier nichts ändern!
checkG(K)

✓ Deine Antwort ist korrekt!
```

Out[7]: 3.870295361503984

Damit wir uns mehr unter dieser Zahl vorstellen können, stellen wir ein paar Vergleiche zu anderen Sprachmodellen an.

2) Vergleich der Modelle

Wir wollen unser Modell nun mit einer rein zufälligen Auswahl eines Folgewortes aus dem Vokabular, also aus allen unterschiedlichen Wörtern, die im Test auftreten, vergleichen. Jedes Wort ist somit gleich wahrscheinlich. Ist unser Modell besser als der Zufall?



a) Wahrscheinlichkeit für die zufällige Wortauswahl

Bestimme die Wahrscheinlichkeit, mit der ein Wort rein zufällig aus dem Vokabular ausgewählt wird.

▼ Lösung (Erst die Frage beantworten, dann hier klicken!)

Da die Auswahl rein zufällig geschieht und deshalb jedem Wort die gleiche Wahrscheinlichkeit zugeordnet wird,

handelt es sich um eine Gleichverteilung. Es gibt V Wörter, auf die wir die Wahrscheinlichkeit von eins gleichmäßig aufteilen müssen. Die Wahrscheinlichkeit ergibt sich somit zu

$$P_Z = \frac{1}{V}.$$



b) Vergleich mit einer zufälligen Wortauswahl

Berechne nun die Kreuzentropie des Zufall-Modells. Mit geschickten Umformungen ist hier kaum Rechenaufwand nötig. Mache dir am besten Notizen zu deinen Umformungen.

☞ Wir nehmen an, dass die Vokabulargröße V der Zahl 75000 entspricht, welche als Größe des deutschen Standardwortschatzes gilt.

▼ ☰ Tipp 1

Die Modellwahrscheinlichkeit ergibt sich als das Produkt der Übergangswahrscheinlichkeiten zu

$$\mathbb{P}(\text{Wort } 1, \dots, \text{Wort } n) = \mathbb{P}(\text{Wort } 1) \cdots \mathbb{P}(\text{Wort } n) = \mathbb{P}(\text{Wort } 1)^n = P_Z^n = \left(\frac{1}{V}\right)^n.$$

▼ ☰ Tipp 2

Die Kreuzentropie ergibt sich somit zu

$$K = -\frac{1}{n} \cdot \log\left(\left(\frac{1}{V}\right)^n\right).$$

Wie kannst du diesen Ausdruck vereinfachen?

▼ ☰ Tipp 3

Erinnere dich an die Logarithmusgesetze. [Hier](#) kannst du sie nochmal nachlesen.

▼ ☰ Tipp 4

Durch geschicktes Umformen ergibt sich die Kreuzentropie zu

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort } 1, \dots, \text{Wort } n)) = -\frac{1}{n} \cdot \log\left(\left(\frac{1}{V}\right)^n\right) = -\frac{1}{n} \cdot n \cdot \log\left(\frac{1}{V}\right) = -\log\left(\frac{1}{V}\right) = \log(V).$$

Wir erkennen, dass die Kreuzentropie des Zufallsmodells gerade dem Logarithmus der Vokabulargröße entspricht.

In [8]: # Ersetze NaN

```
K_Z = #= NaN =# log(75000)
```

```
# Hier nichts ändern!
checkH(K_Z)
```

✓ Deine Antwort ist korrekt!

Damit ist die Kreuzentropie des Mittelwert-Modells wesentlich kleiner als die Kreuzentropie des Zufalls-Modells.

Out[8]: 11.225243392518447



c) Kreuzentropie verschiedener N -Gramm-Modelle

Wir wollen die einzelnen N -Gramm-Modelle miteinander vergleichen. Berechne die Kreuzentropie für Uni-, Bi- und Tri-Gramm-Modell mit Glättung. Gewichte dazu die Glättungswahrscheinlichkeit mit dem Faktor 0.1 und die jeweilige N -Gramm-Wahrscheinlichkeit mit dem Faktor 0.9 (Achtung: Die Glättung ist notwendig, da wir sonst wieder das Problem mit einer Modellwahrscheinlichkeit von null haben).

 Nutze für die Berechnung der Kreuzentropie die Funktion `KE(g_0,g_1,g_2,g_3)`. Ersetze dazu die Werte `g_0,g_1,g_2` bzw. `g_3` durch die Wahl deiner Gewichte, also z.B. `KE(3/18,2/18,5/18,8/18)`.

```
In [9]: # Kreuzentropie Uni_Gramm & Glättung  
K_Uni = KE(NaN,NaN,NaN,NaN) =# KE(0.1,0.9,0,0)
```

```
Out[9]: 7.89484276558931
```

```
In [10]: # Kreuzentropie Bi_Gramm & Glättung  
K_Bi = KE(NaN,NaN,NaN,NaN) =# KE(0.1,0,0.9,0)
```

```
Out[10]: 5.960639671804403
```

```
In [11]: # Kreuzentropie Tri_Gramm & Glättung  
K_Tri = KE(NaN,NaN,NaN,NaN) =# KE(0.1,0,0,0.9)
```

```
Out[11]: 5.34646715709661
```



d) Vergleich der N -Gramm-Modelle

Interpretiere die Ergebnisse im Hinblick auf den Vergleich der Qualität der drei Sprachmodelle.



e) Kreuzentropie für ein kombiniertes Modell mit den Gewichten deiner Wahl

Lege nun auf Basis deiner Überlegungen die Gewichte fest und berechne für diese Gewichtung der N -Gramme die Kreuzentropie. Du darfst natürlich auch mehrere Möglichkeiten ausprobieren.

```
In [12]: # Kreuzentropie für die Gewichtung deiner Wahl  
K_Test = KE(NaN,NaN,NaN,NaN) =# KE(0.25,0.25,0.25,0.25)
```

```
Out[12]: 4.053117689076634
```

Schon fertig?

Was passiert, wenn wir einen anderen Trainingsdatensatz verwenden und unser Modell zum Beispiel mit Texten von Johann Wolfgang von Goethe trainieren? Auf diesem [Zusatzblatt](#) hast du die Möglichkeit, den Einfluss der Trainingsdaten auf das Modell zu untersuchen.

Diskussion im Plenum

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!

Anschließend geht es weiter mit [Arbeitsblatt 5](#) oder [Arbeitsblatt 5 open](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"2.

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.5 Arbeitsblatt 5

Arbeitsblatt 5 | Bestimmung der optimalen Gewichte



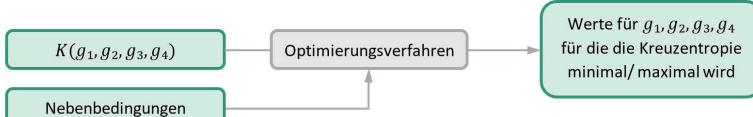
Wir wollen nun die **Kreuzentropie** nutzen, um die **optimalen Gewichte** für den gewichteten Mittelwert der Übergangswahrscheinlichkeiten zu finden.

Erinnerung: Eine kleine Kreuzentropie bedeutet, dass das Sprachmodell zu unserem Testdatensatz passt und den Sprachgebrauch für diese Art von Texten gut modelliert.

Diejenigen Gewichte, die die Kreuzentropie minimieren, passen das Modell somit am besten an den Testdatensatz an. Wir müssen daher das Minimum der Kreuzentropie in Abhängigkeit der Gewichte sowie die zugehörigen Gewichte bestimmen.

Optimierungsproblem

Hierbei handelt es sich um ein sogenanntes **Optimierungsproblem**, bei dem das Minimum bzw. Maximum einer Funktion, in unserem Fall das Minimum der Kreuzentropie $K(g_0, g_1, g_2, g_3)$, gesucht ist. Diese Funktion wird auch als **Zielfunktion** bezeichnet. Gesucht werden diejenigen Werte der Variablen, für die die Zielfunktion maximal oder minimal wird. In unserem Fall sind die Gewichte g_0, g_1, g_2 und g_3 gesucht, für die die Kreuzentropie minimal wird. In den meisten Fällen, wie auch bei uns, sollen diese Variablen bestimmte **Nebenbedingungen** erfüllen.



Die typische Strategie, die Zielfunktion abzuleiten und null zu setzen, um das Minimum zu bestimmen, funktioniert hier allerdings nicht. Eine Schwierigkeit ist, dass die Zielfunktion eine Funktion von mehreren Veränderlichen ist. Sie hängt nicht nur von einem, sondern gleich vier Gewichten ab. Zum anderen hat die Zielfunktion eine ziemlich komplexe Gestalt. Es gibt allerdings sogenannte **Optimierungsverfahren**, die sich mit Hilfe einer festgelegten Systematik schrittweise dem Minimum nähern.

Wir überlassen die Wahl des Optimierungsverfahrens und das Durchführen des Optimierungsverfahrens dem Computer.



a) Anwendung eines Optimierungsverfahrens

Trage

- die Zielfunktion
- sowie alle notwendigen Nebenbedingungen, die für die Gewichte gelten sollen,

in das untenstehende Codefeld ein. Führe das Codefeld anschließend aus, um dir das über das Optimierungsverfahren bestimmte Minimum der Zielfunktion sowie die zugehörigen Gewichte ausgeben zu lassen.

**Variablen und Schreibweise:**

- Mehrere Nebenbedingungen trennst du mit `&&`.
- Nebenbedingungen mit einem Gleichheitszeichen müssen mit einem doppelten Gleichheitszeichen `==` notiert werden. Die Programmiersprache Julia nutzt dies zur Unterscheidung zwischen Definitionen und Gleichungen.
- Auf die Kreuzentropie kannst du über die implementierte Funktion `KE(g_0,g_1,g_2,g_3)` zugreifen.

```
In [1]: #Hier nichts ändern!
include("../code/checkAB5_short.jl");

# Ersetze NaN

Zielfunktion(g_0,g_1,g_2,g_3) = ## NaN; =# KE(g_0,g_1,g_2,g_3)
Nebenbedingung(g_0,g_1,g_2,g_3) = ## NaN; =# g_0*g_1+g_2+g_3==1 && g_0>=0 && g_1>=0 && g_2>=0 && g_3>=0

#Hier nichts ändern!
checkAB5_short(Zielfunktion,Nebenbedingung)
```

Deine Zielfunktion, sowie die Nebenbedingungen wurden korrekt aufgestellt.

Die Gewichte ergeben sich zu

$g_0 = 0.04, g_1 = 0.20, g_2 = 0.14, g_3 = 0.62$

mit einer Kreuzentropie von 3.76.

**b) Interpretation der Lösung des Optimierungsverfahrens**

Interpretiere das Ergebnis und vergleiche es mit deinen intuitiven Überlegungen auf [Arbeitsblatt 3 Aufgabe 1d](#)).

Schon fertig?**Zusatz**

Du bist schon fertig? Dann hast du nun die Möglichkeit,

... dein Modell wie bei ChatGPT zum Generieren von Texten zu nutzen. [Hier](#) geht es zum zugehörigen Arbeitsblatt.

... schon mal weiter zum [Arbeitsblatt Diskussion](#) zu gehen und dir Gedanken zu möglichen Nachteilen von Assistenzsystemen wie den Wortvorschlägen zu machen.

Diskussion im Plenum 

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen

4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"¹ und den Texten der Kategorie "Belletistik" des Korpus "LIMAS"².

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.6 Arbeitsblatt 5 open

Arbeitsblatt 5 open | Bestimmung der optimalen Gewichte



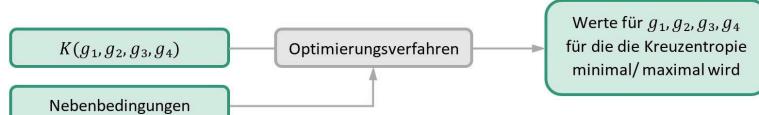
Wir wollen nun die **Kreuzentropie** nutzen, um die **optimalen Gewichte** für den gewichteten Mittelwert der Übergangswahrscheinlichkeiten zu finden.

Erinnerung: Eine kleine Kreuzentropie bedeutet, dass das Sprachmodell zu unserem Testdatensatz passt und somit das Sprachmodell für diese Art von Texten den Sprachgebrauch gut modelliert.

Diejenigen Gewichte, die die Kreuzentropie minimieren, passen das Modell somit am besten an den Testdatensatz an. Wir müssen daher das Minimum der Kreuzentropie in Abhängigkeit der Gewichte finden.

Optimierungsproblem

Hierbei handelt es sich um ein sogenanntes **Optimierungsproblem**, bei dem das Minimum bzw. Maximum einer Funktion, in unserem Fall das Minimum der Kreuzentropie $K(g_0, g_1, g_2, g_3)$, gesucht ist. Diese Funktion wird auch als **Zielfunktion** bezeichnet. Gesucht werden diejenigen Werte der Variablen, für die die Zielfunktion maximal oder minimal wird. In unserem Fall sind die Gewichte g_0, g_1, g_2 und g_3 gesucht, für die die Kreuzfunktion minimal wird. In den meisten Fällen, wie auch bei uns, sollen diese Variablen bestimmte **Nebenbedingungen** erfüllen.



Die typische Strategie, die Zielfunktion abzuleiten und null zu setzen, um das Minimum zu bestimmen, funktioniert hier allerdings nicht. Eine Schwierigkeit ist, dass die Zielfunktion eine Funktion von mehreren Veränderlichen ist. Sie hängt nicht nur von einem, sondern gleich vier Gewichten ab. Zudem hat die Zielfunktion

$$K(g_0, g_1, g_2, g_3) = -\frac{1}{n} \cdot (\log(g_0 \cdot \frac{1}{V} + g_1 \cdot P_{Uni}(\text{Wort 1}) + g_2 \cdot P_{Bi}(\text{Wort 1}) + g_3 \cdot P_{Tri}(\text{Wort 1})) + \dots + \log(g_0 \cdot \frac{1}{V} + g_1 \cdot P_{Uni}(\text{Wort } n) + g_2 \cdot P_{Bi}(\text{Wort } n) + g_3 \cdot P_{Tri}(\text{Wort } n)))$$

eine ziemlich komplexe Gestalt. Wir müssen daher einen anderen Weg finden, um die Kreuzentropie in Abhängigkeit der Gewichte zu minimieren.



a) Entwicklung eines Optimierungsverfahrens

Überlege dir ein geeignetes Verfahren, um diejenigen Gewichte zu bestimmen, die die Kreuzentropie des Testdatensatzes $K(g_0, g_1, g_2, g_3)$ minimieren.

Variablen: Auf die Kreuzentropie kannst du über die implementierte Funktion `KE(g_0, g_1, g_2, g_3)` zugreifen.

Tipp 1

Überlege dir, welche Nebenbedingungen für die Gewichte gelten und wie du diese bei der Minimierung berücksichtigen kannst.

▼  **Tipp 2**

Auf Arbeitsblatt 4 hast du bereits durch Ausprobieren verschiedene Gewichtungen und deren Kreuzentropie verglichen. Kannst du dieses Ausprobieren systematisieren?

▼  **Tipp 3**

Frage deine Betreuenden :).

```
In [1]: #Hier nichts ändern!
include("../code/checkAB5_short.jl");
```

```
In [2]: # Hier ist Platz für deinen Code!
```



b)

Interpretiere das Ergebnis und vergleiche es mit deinen intuitiven Überlegungen auf [Arbeitsblatt 3 Aufgabe 1d](#)).

Schon fertig?

Zusatz

Du bist schon fertig? Dann hast du nun die Möglichkeit,

... dein Modell wie bei ChatGPT zur Generierung von ganzen Texten zu nutzen. [Hier](#) geht es zum zugehörigen Arbeitsblatt.

... den Einfluss der Trainingsdaten näher zu untersuchen. [Hier](#) geht es zum zugehörigen Arbeitsblatt.

Diskussion im Plenum

Wir werden deine bisherigen Ergebnisse gemeinsam diskutieren!



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"2.
Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.7 Zusatzblatt 3

Zusatzblatt 3 | Die Rückfallstrategie



Teil 1) Verknüpfung der N -Gramm-Modelle mit einer Rückfallstrategie

Eine weitere Möglichkeit zur Kombination von N -Gramm-Modellen besteht darin, sie durch eine sogenannte **Rückfallstrategie** zu verknüpfen. Stellen wir uns vor, wir arbeiten mit einem Tri-Gramm-Modell. Liegt nun eine Wortvorgeschichte vor, die nicht im Datensatz auftritt, so können wir mithilfe dieses Modells keinen Vorschlag machen.



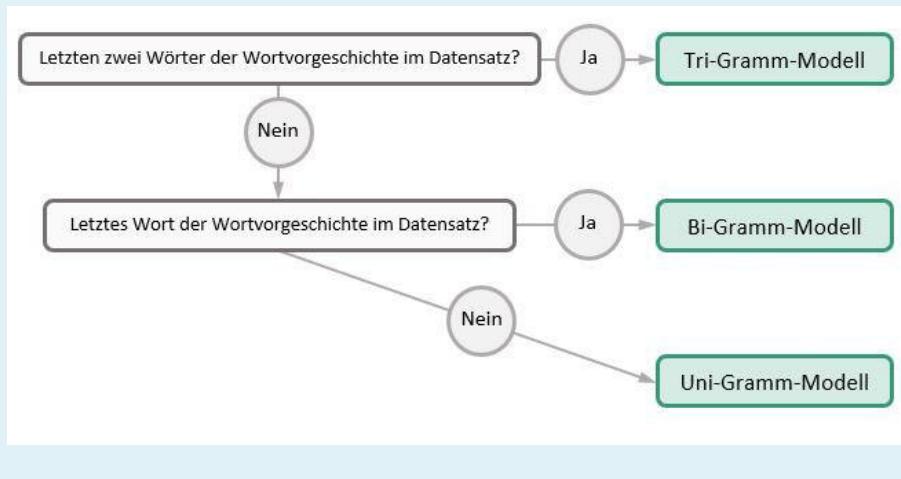
a) Die Rückfallstrategie

Gibt es unter Einbezug der anderen N -Gramm-Modelle eine Möglichkeit, dennoch ein Wort vorzuschlagen? Formuliere eine allgemeine Strategie, mit der es möglich ist, die Wortvorgeschichte so lange wie möglich zu wählen und dennoch immer ein Vorschlag zu generieren.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Im Falle eines ungewöhnlichen Bi-Gramms als Wortvorgeschichte gibt uns das Tri-Gramm-Modell keine Vorschläge für das nächste Wort. Wir ziehen daher das Bi-Gramm-Modell heran. Tritt auch das letzte Wort der Wortvorgeschichte nicht im Trainingsdatensatz auf, so verwenden wir das Uni-Gramm-Modell und schlagen einfach die Wörter vor, die am häufigsten im Datensatz auftreten. Dieses Verfahren wird als **Rückfallstrategie** bezeichnet.

Die Strategie können wir folgendermaßen veranschaulichen:



b) Rückfallstrategie an einem Beispiel

Diese Rückfallstrategie wollen wir nun umsetzen. Wir gehen davon aus, dass bereits die Wörter "ich würde gerne" eingetippt wurden. Das nächste Wort soll mithilfe der Rückfallstrategie vorgeschlagen werden.

Ersetze in den untenstehenden Codefeldern die `NaN` so, dass die **if-Anweisung** die Rückfallstrategie umsetzt und ein Vorschlag für das nächste Wort gemacht wird. Verwende hierzu:

- die Tabellen der relativen Häufigkeiten der Uni- (`T_Uni`), Bi (`T_Bi`) und Tri-Gramme (`T_Tri`).
- die in der Tabelle angegebenen Befehle (die **blauen** Begriffe zeigen an, dass hier die passende Größe ersetzt werden muss; die Wortvorgeschichte muss mit `" "` eingegeben werden):

Befehl	Bedeutung
<code>bestesWort_Uni(w)</code>	Generiert Wortvorschläge für die eingegebene Wortvorgeschichte <code>w</code> auf Basis des Uni-Gramm-Modells
<code>bestesWort_Bi(w)</code>	Generiert Wortvorschläge für die eingegebene Wortvorgeschichte <code>w</code> auf Basis des Bi-Gramm-Modells
<code>bestesWort_Tri(w)</code>	Generiert Wortvorschläge für die eingegebene Wortvorgeschichte <code>w</code> auf Basis des Tri-Gramm-Modells
<code>w in find(T)</code>	Überprüft, ob die Wortvorgeschichte <code>w</code> in der Tabelle <code>T</code> enthalten ist

 Falls du if-Anweisungen noch nicht kennst, kannst du dir [hier](#) die Information zu if-Anweisungen ansehen.

Tipp

```
In [1]: # Hier nichts ändern
include("../code/checkZ3.jl");
function checkif(T_Tri,T_Bi)

# Ersetze NaN

if  == NaN == "würde gerne" in find(T_Tri) # Bedingung I
    Vorschlag = == NaN == bestesWort_Tri("würde gerne") # Anweisung I
elseif == NaN == "gerne" in find(T_Bi) # Bedingung II
    Vorschlag = == NaN == bestesWort_Bi("gerne") # Anweisung II
else
    Vorschlag = == NaN == bestesWort_Uni("gerne") # Anweisung III
end

# Hier nichts ändern
Vorschlag
end
checkA(checkif(["würde gerne"],[""]),
      checkif([""],["gerne"]),
      checkif([""],[""]))

✓ Deine Antwort ist korrekt!
```

Out[1]: 2-element Vector{String}:
"für"
"einfach"

Zusatz 1) Rückfallstrategie im allgemeinen Fall

Auf diesem [Zusatzblatt](#) kannst du die Rückfallstrategie für den allgemeinen Fall implementieren.

Zusatz 2) Garantiere, dass drei Wörter vorgeschlagen werden!

Wie du vielleicht bemerkt hast, ist es bei unserer Rückfallstrategie möglich, dass weniger als drei Wortvorschläge gegeben werden. Auf diesem [Zusatzblatt](#) verbesserst du das Modell, sodass immer drei Vorschläge ausgegeben werden.

Teil 2) Vergleich beider Kombinationsmöglichkeiten

Wir werden in den folgenden Arbeitsblättern trotz der einfachen Herangehensweise, die Modelle mithilfe der Rückfallstrategie zu kombinieren, mit dem Mittelwertmodell weiterarbeiten. Ein Grund hierfür ist der folgende: Da insbesondere die relativen Häufigkeiten der Tri-Gramme und teilweise auch der Bi-Gramme oft auf sehr wenigen Beobachtungen basieren, besteht die Gefahr, dass sie den tatsächlichen Sprachgebrauch nicht zuverlässig widerspiegeln. Betrachten wir hierzu ein kleines Beispiel: Das Tri-Gramm-Modell liefert zwei mögliche Wortvorschläge für die Wortvorgeschichte „würde gerne“: „für“ und „einfach“. Es kann jedoch vorkommen, dass die entsprechenden Tri-Gramme „würde gerne für“ und „würde gerne einfach“ im Datensatz nur ein- oder zweimal auftreten. In diesem Fall basieren unsere Vorschläge auf sehr geringen Anzahlen und spiegeln möglicherweise nicht den typischen Satzverlauf für die gegebene Wortvorgeschichte wider. Daher wäre es sinnvoll, zusätzlich die Vorschläge des Bi-Gramm-Modells zu betrachten. Zählen die Vorschläge des Tri-Gramm-Modells auch hier zu den besten Vorschlägen dazu, so können wir uns sicher sein, dass es sich bei den Wortvorschlägen aus dem Tri-Gramm-Modell tatsächlich um Wörter handelt, die in diesem Kontext häufiger verwendet werden. Andernfalls sollte abgewogen werden, ob nicht ein vom Bi-Gramm-Modell vorgeschlagenes Wort bevorzugt werden sollte. Es ist also sinnvoll, die relativen Häufigkeiten der Bi- und Tri-Gramme bei der Bestimmung eines Wortvorschages mit einzubeziehen. Auch die Information, wie häufig ein Wort im Allgemeinen auftritt, kann wichtig für die Wahl des Wortvorschages sein. Beim Mittelwert Modell werden alle der genannten Informationen genutzt. Trotzdem ist es sinnvoll, auch die Rückfallstrategie zu kennen. In der Praxis werden häufig kompliziertere Techniken verwendet, die jedoch auf beider hier vorgestellten Varianten basieren.

Hier geht es zurück zu [Arbeitsblatt 3](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"2.

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.8 Zusatzblatt 4

Zusatzblatt 4 | Der Einfluss der Trainingsdaten



Auf diesem Zusatzblatt wollen wir den **Einfluss der Trainingsdaten auf das Sprachmodell** untersuchen. Der bislang verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland", einer Sammlung von WhatsApp Mitteilungen und den Texten der Kategorie "Belletistik", also Unterhaltungsliteratur, wie beispielsweise Romane oder Krimis, des Korpus "LIMAS". Wir bezeichnen diesen Datensatz im weiteren Verlauf als Datensatz 1. Wir schauen uns jetzt an, was passiert, wenn wir Werke von Johann Wolfgang von Goethe, einem deutschen Schriftsteller, der von 1749 bis 1832 lebte, als Trainingsdaten nutzen.



a) Vergleich der Wortvorschläge

Wir wollen nun das mit den Texten von Goethe trainierte Modell mit einigen Wörtern testen und die erhaltenen Vorschläge mit den Vorschlägen, die man mit dem ursprünglichen Modell erhält, vergleichen. Setze dazu verschiedene Wortvorgeschichten für `NaN` ein (z.B. "guten", "nicht ich", "ich brauche", "mir dieses"). Was stellst du fest, wenn du die Vorschläge vergleichst?

In [1]: `# Ersetze NaN`

```
Wortvorgeschichte = #= "NaN"; =# "guten"
#Hier nichts ändern!
include("../code/checkZ4.jl"); checkA(Wortvorgeschichte)
```

Vorschläge des Modells trainiert mit Datensatz 1 Vorschläge des Modells trainiert mit Datensatz Goethe

morgen	kindes
start	tage
tag	menschen



b) Vergleich der Kreuzentropie

Diesen Vergleich können wir quantifizieren, indem wir die Bewertung beider Modelle betrachten. Führe das untenstehende Codefeld aus, um die Kreuzentropie beider Modelle zu berechnen. Die Kreuzentropie wird auf Basis des ursprünglichen, aus einzelnen Chatnachrichten und Buchtexten der letzten Jahrzehnte bestehenden Testdatensatzes, welchen wir als repräsentativ für die Alltagssprache erachten, bestimmt. Vergleiche anschließend die Ergebnisse. Bestätigen sich deine Überlegungen aus Teilaufgabe a)?

In [2]: `#Hier nichts ändern!`

```
K1 = KE(1/4, 1/4, 1/4, 1/4) #Kreuzentropie des Modells trainiert mit Datensatz 1
K2 = KE_g(1/4, 1/4, 1/4, 1/4) #Kreuzentropie des Modells trainiert mit dem Datensatz Goethe
checkB(K1,K2)
```

Kreuzentropie des Modells trainiert mit Datensatz 1 Kreuzentropie des Modells trainiert mit Datensatz Goethe

4.05312

8.33574



c) Folgerung für die Auswahl des Trainingsdatensatzes

Formuliere allgemeine Anforderungen an die Eigenschaften eines geeigneten Trainingsdatensatzes.

▼ **Lösung** (Erst die Frage beantworten, dann hier klicken!)

Unser Modell nutzt die relativen Häufigkeiten der N -Gramme aus den Trainingsdaten, um Vorschläge zu generieren. Ein maschinelles Lernalgorithmus kann deshalb nur so gut sein wie seine Trainingsdaten! Um gute Vorhersagen treffen zu können, ist es daher entscheidend, dass die Trainingsdaten zur Anwendung, in der das Sprachmodell verwendet wird, gut passen und die zu verallgemeinernden neuen Situationen hinsichtlich Vokabular und Textart angemessen repräsentieren.
In unserem Fall sollten die Trainingstexte daher nicht aus dem 17. Jahrhundert stammen, da in dieser Zeit andere Wörter verwendet wurden als in der heutigen Gegenwartssprache. Dies könnte dazu führen, dass für unsere Anwendung weniger passende Vorschläge generiert werden.



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der große Datensatz Datensatz_LW besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletristik" des Korpus "LIMAS"2.

Der Datensatz Goethe besteht aus den Werken Johann Wolfgang von Goethes: "Die Leiden des jungen Werthers" Band 13 und Band 24, "Die Wahlverwandtschaften" Band 15, "Wilhelm Meisters Lehrjahre" Band 46.
Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

3 Goethe, Johann Wolfgang von: Die Leiden des jungen Werthers. Bd. 1. Leipzig, 1774. In: Deutsches Textarchiv https://www.deutschestextarchiv.de/goethe_werther01_1774

4 Goethe, Johann Wolfgang von: Die Leiden des jungen Werthers. Bd. 2. Leipzig, 1774. In: Deutsches Textarchiv https://www.deutschestextarchiv.de/goethe_werther02_1774

5 Goethe, Johann Wolfgang von: Die Wahlverwandtschaften. Bd. 1. Tübingen, 1809. In: Deutsches Textarchiv https://www.deutschestextarchiv.de/goethe_wahlverw01_1809

6 Goethe, Johann Wolfgang von: Wilhelm Meisters Lehrjahre. Bd. 4. Frankfurt (Main) u. a., 1796. In: Deutsches Textarchiv https://www.deutschestextarchiv.de/goethe_lehrjahre04_1796

A.9 Zusatzblatt 5

Zusatzblatt 5 | Texte erzeugen



Bisher haben wir mithilfe unseres Modells nur Vorschläge für das nächste Wort abgegeben. Könnte man die entwickelten Modelle nutzen, um stattdessen einen ganzen Text zu generieren?

In [1]: `#Hier nichts ändern!
include("../code/checkAB_open.jl");`



a) Texte generieren

Werde kreativ! Überlege dir ein Verfahren, mit dem Texte generiert werden können. Nutze dazu die von uns entwickelten Modelle in den vorangegangenen Arbeitsblättern.

🔧 Variablen:

- Auf die generierten Wortvorschläge auf Basis des Uni-Gramm-, Bi-Gramm- und Tri-Gramm-Modells kannst du mit `bestesWort_Uni(w)`, bzw. `bestesWort_Bi(w)` und `bestesWort_Tri(w)` zugreifen.
- Auf die generierten Wortvorschläge auf Basis des Rückfall-Modells kannst du mit `bestesWort_rückfallZ(w)` zugreifen.
- Auf die generierten Wortvorschläge auf Basis des Mittelwert-Modells kannst du mit `bestesWort_MW(w,g_0,g_1,g_2,g_3)` zugreifen.

🔧 Hier kannst du dir nach Bedarf nochmal die Information zu if-Anweisungen und [hier](#) die Information zu for-Schleifen ansehen.

▼ ⚡ Tipp 1

Wähle an deinem Handy mehrmals hintereinander den mittleren Wortvorschlag aus. Überlege dir, wie du diesen Vorgang im Code umsetzen kannst.

▼ ⚡ Tipp 2

Teste verschiedene Modelle aus (Uni-Gramm-, Bi-Gramm-, Tri-Gramm-, Rückfall- und Mittelwert-Modell). Überprüfe welches Modell die besten Ergebnisse liefert.

▼ ⚡ Tipp 3

Überlege, wie du deine Texte menschenähnlicher gestalten kannst. Du kannst beispielsweise Endlosschleifen unterbinden. Diese entstehen, wenn ein Wort ein zweites Mal auftritt. Der nachfolgende erzeugte Text gleicht dem Text nach dem ersten Auftreten des Wortes. Hast du eine Idee, wie du deinen Code verändern kannst, sodass der Text abwechslungsreicher wird?

In [2]: `# Hier ist Platz für deinen Code`

Lösungsvorschlag:

In [3]: `w = ["Es", "war", "einmal"]
for i in 1:100`

```
w = push!(w,bestesWort_rückfallZ(string(w[i+1]," ",w[i+2]))[rand((1,2))])  
end  
  
print(w)  
  
["Es", "war", "einmal", "ein", "schriftsteller", "der", "mochte", "tiere", "so", "ein", "bisschen", "was", "besonderes", "solltest", "du", "schon", "mal", "anfangen", "meine", "mutter", "hat", "gefragt", "ob", "ich", "das", "nicht", "toll", "hab", "eben", "auch", "gedacht", "und", "ich", "hab", "den", "kerl", "mit", "seinen", "fäusten", "hinauszuprügeln", "ich", "hab", "das", "geld", "aus", "aber", "nach", "2", "3", "monaten", "wieder", "in", "die", "schule", "los", "angeles", "tage", "der", "es", "in", "den", "garten", "wir", "haben", "heute", "schon", "die", "2", "tage", "verreisen", "und", "ich", "hab", "den", "kerl", "mit", "unserem", "betrieb", "ist", "ein", "rohling", "ist", "ein", "rohling", "ist", "das", "den", "raus", "finden", "es", "raus", "wenn", "ich", "mich", "nicht", "aus", "den", "augen", "den", "strahl", "verfolgend", "bis", "zum"]
```



b) Probleme und ein Vergleich der Texte

Beschreibe die Probleme, die bei der Textgenerierung auftreten. Vergleiche die generierten Texte, die mit unterschiedlichen Modellen (Uni-/ Bi-/ Tri-Gramm-Modell, Rückfall-Modell, Mittelwert-Modell) entstehen. Welche Probleme treten bei allen Modellen auf, und welche Probleme können bei bestimmten Modellen reduziert oder sogar vollständig gelöst werden?



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Der verwendete Datensatz besteht aus den deutschsprachigen Texten des Korpus "What's up, Switzerland"1 und den Texten der Kategorie "Belletistik" des Korpus "LIMAS"2.

Autorin: Stephanie Hofmann

1 Stark, Elisabeth; Ueberwasser, Simone; Göhring, Anne (2014-2020). Korpus "What's up, Switzerland?". Universität Zürich. www.whatsup-switzerland.ch.

2 Forschungsgruppe LIMAS (1970-1971). LIMAS-KORPUS (Linguistik und Maschinelle Sprachbearbeitung). Universität Bonn, Universität Regensburg.

A.10 Präsentationsfolien

Auf den nachfolgenden Seiten sind die Präsentationsfolien abgedruckt. Alle Fragen, die an die Lernenden gestellt werden, sind mit einem Symbol gekennzeichnet, welches zwei Personen und eine Sprechblase mit einem Fragezeichen darstellt.



Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Einführung in die Problemstellung



Eingabehilfen auf mobilen Endgeräten

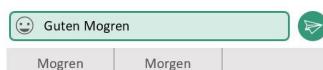
Welche Eingabehilfen am Handy oder Tablet kennt ihr?



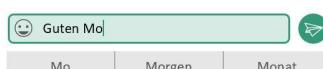
Eingabehilfen auf mobilen Endgeräten

Welche Eingabehilfen am Handy oder Tablet kennt ihr?

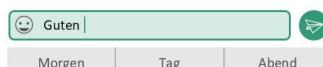
- Autokorrektur



- Wortvervollständigung



- Wortvorschläge



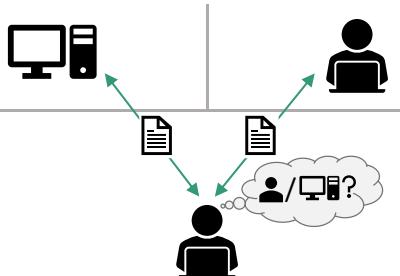
Wozu können Wortvorschläge genutzt werden?



Wozu können Wortvorschläge genutzt werden?

- als **Eingabehilfe** beim Smartphone oder bei der Suchanfrage im Webbrowser
 - Weniger Zeit beim Tippen
 - Weniger Fehler beim Tippen
- für die Spracherkennung
- für die Schrifterkennung
- für das maschinelle Übersetzen
- als Assistenzsystem für Personen mit verbalen und motorischen Schwächen

Was hat das Generieren von Wortvorschlägen mit Künstlicher Intelligenz zu tun?



Turing-Test 1950 von Alan Turing

A Lernmaterial zum Thema Wortvorhersage

Was hat das Generieren von Wortvorschlägen mit Künstlicher Intelligenz zu tun?



Was verbindet ihr mit dem Begriff KI?

Welche Anwendungen der KI kennt ihr?

 CAMMP Workshop | Wortvorschläge | 7

Wortvorschläge | ein Beispiel

Ich mag Physik. Ich mag Sport. Ich liebe Mathe.




 CAMMP Workshop | Wortvorschläge | 8

Wortvorschläge | ein Beispiel

Ich mag Physik. Ich mag Sport. Ich liebe Mathe.




 CAMMP Workshop | Wortvorschläge | 9

Wortvorschläge | ein Beispiel

Ich mag Physik. Ich mag Sport. Ich liebe Mathe.




 CAMMP Workshop | Wortvorschläge | 10

Wortvorschläge | ein Beispiel

Ich mag Physik. Ich mag Sport. Ich liebe Mathe.




 CAMMP Workshop | Wortvorschläge | 11

Maschinelles Lernen

Regelbasiertes Verfahren

Feste Regeln

z.B. schlage nach dem Wort "Guten" die Wörter "Tag" und "Morgen" vor
→ eine solche Regel müsste für jedes Wort aufgestellt werden

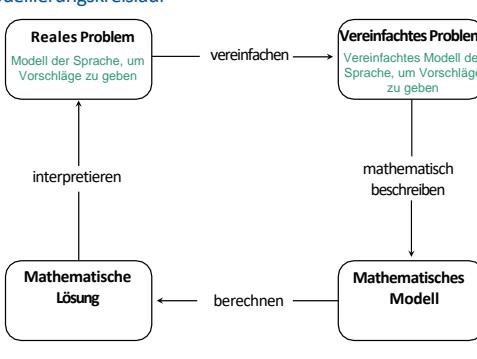
Maschinelles Lernen

Lernen aus Daten

z.B. Häufigkeit, mit der ein Wort nach den geschriebenen Wörtern auftritt, aus gegebenen Textdaten bestimmen

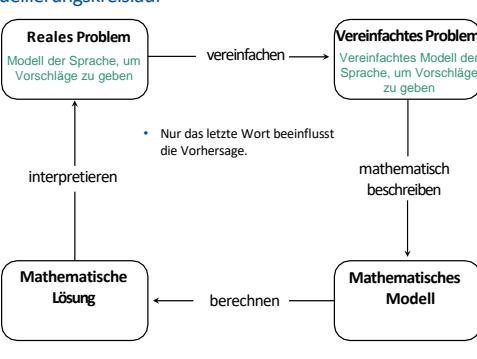
 CAMMP Workshop | Wortvorschläge | 12

Modellierungskreislauf



 CAMMP Workshop | Wortvorschläge | 13

Modellierungskreislauf



 CAMMP Workshop | Wortvorschläge | 14

Ziel und Ablauf

Ziel: Entwicklung eines eigenen Sprachmodells zur Wortvorhersage!

Ablauf:

1. Ein Sprachmodell auf einem kleinen Trainingsdatensatz entwickeln.
2. Die Erstellung des Sprachmodells automatisieren. Das Sprachmodell auf einem größeren Datensatz anwenden und stetig verbessern!
3. Das entwickelte Sprachmodell bewerten.
4. Kritisch reflektieren: Was sollte bei der Entwicklung und Nutzung von Sprachmodellen kritisch reflektiert werden?

Jetzt seid ihr dran ...

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuenden!

Schritte zum Arbeitsmaterial

1. Öffne workshops.cammp.online
2. Auf „Zugriff auf Lernmaterial“ und dann auf „Registrieren!“ klicken
3. Account erstellen: der Username muss Präfix **cammp_** enthalten (z. B. **cammp_laura1234**)
4. Auf „Anmelden!“ klicken, Accountdaten eingeben und einloggen
5. Öffne die Datei **Willkommen_CAMMP**
6. Im Dropdown Menü **Wortvorschläge** auswählen und herunterladen
7. Ordner **wortvorschlaege** und dann Ordner **worksheets** öffnen
8. Los geht's mit Arbeitsblatt 1!

Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Diskussion nach AB 1

Modellierungskreislauf

```

graph TD
    A[Reales Problem  
Modell der Sprache, um Vorschläge zu geben] -- vereinfachen --> B[Vereinfachtes Problem  
Vereinfachtes Modell der Sprache, um Vorschläge zu geben]
    B -- mathematisch beschreiben --> C[Mathematisches Modell  
?]
    C -- berechnen --> D[Mathematische Lösung]
    D -- interpretieren --> A
    
```

Modellierungskreislauf

```

graph TD
    A[Reales Problem  
Modell der Sprache, um Vorschläge zu geben] -- vereinfachen --> B[Vereinfachtes Problem  
Vereinfachtes Modell der Sprache, um Vorschläge zu geben]
    B -- mathematisch beschreiben --> C[Mathematisches Modell]
    C -- berechnen --> D[Mathematische Lösung]
    D -- interpretieren --> A
    
```

- Nur das letzte Wort beeinflusst die Vorhersage.
- Die Groß- und Kleinschreibung wird nicht beachtet.
- Der Alltagswortschatz ist statisch und die Trainingstexte repräsentieren diesen ausreichend gut.

Ein erstes Sprachmodell

Ich **mag** Physik. **Ich** **mag** Sport. **Ich** **liebe** Mathe.

$h_{Ich}(mag) =$

Ein erstes Sprachmodell

Ich **mag** Physik. **Ich** **mag** Sport. **Ich** **liebe** Mathe.

$$h_{Ich}(mag) = \frac{N(Ich, \ mag)}{N(Ich, \ Ich) + N(Ich, \ mag) + N(Ich, \ liebe) + \dots}$$

$$= \frac{2}{0 + 2 + 1 + 0 + 0 + 0} = \boxed{\frac{2}{3}}$$

A Lernmaterial zum Thema Wortvorhersage

Ein erstes Sprachmodell

Mithilfe von Übergangstabellen...

? 

Wort2							
		Ich	liebe	mag	Mathe	Physik	Sport
Wort1	Ich	0	$\frac{1}{3}$	$\frac{2}{3}$	0	0	0
	liebe	0	0	0	1	0	0
	mag	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
	Mathe	0	0	0	0	0	0
	Physik	0	0	0	0	0	0
	Sport	0	0	0	0	0	0

CAMMP Workshop | Wortvorschläge | 23

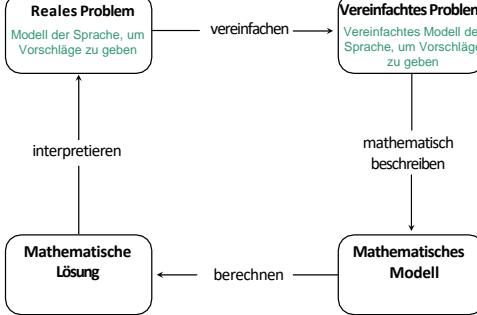
Ein erstes Sprachmodell

Mithilfe von Übergangstabellen...

Wort2							
		Ich	liebe	mag	Mathe	Physik	Sport
Wort1	Ich	0	$\frac{1}{3}$	$\frac{2}{3}$	0	0	0
	liebe	0	0	0	1	0	0
	mag	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
	Mathe	0	0	0	0	0	0
	Physik	0	0	0	0	0	0
	Sport	0	0	0	0	0	0

CAMMP Workshop | Wortvorschläge | 24

Modellierungskreislauf



CAMMP Workshop | Wortvorschläge | 25

Ziel und Ablauf

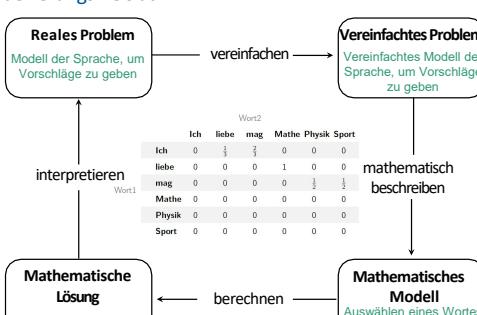
Ziel: Entwicklung eines eigenen Spracherkennungssystems!

Ablauf:

1. Besondere Merkmale aus dem Sprachsignal ermitteln. 
2. Ermittelte Merkmale auf die wichtigsten reduzieren und zu einem Muster zusammenfassen. 
3. Muster der Sprachsignale vergleichen. 
4. Das entwickelte Modell bewerten. 
5. Das Modell stetig verbessern! 

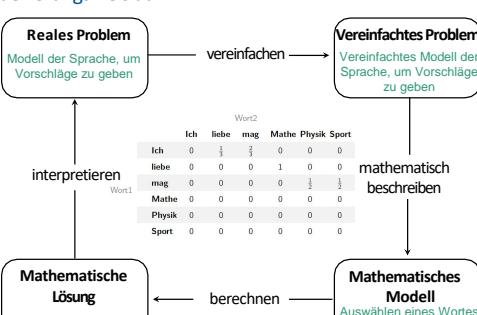
CAMMP Workshop | Spracherkennung | 26

Modellierungskreislauf



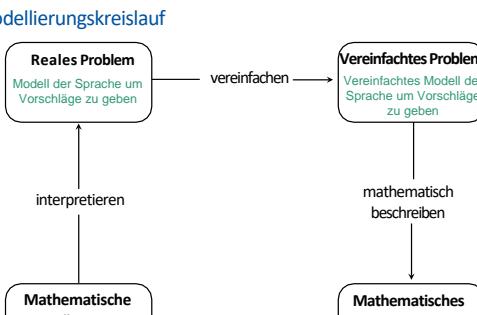
CAMMP Workshop | Wortvorschläge | 27

Modellierungskreislauf



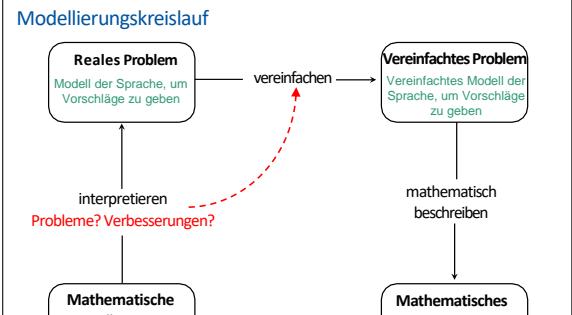
CAMMP Workshop | Wortvorschläge | 28

Modellierungskreislauf



CAMMP Workshop | Wortvorschläge | 29

Modellierungskreislauf



CAMMP Workshop | Wortvorschläge | 30

Ein erstes Sprachmodell

Welche Probleme konntet ihr feststellen?



CAMMP Workshop | Wortvorschläge | 31

Ein erstes Sprachmodell

Welche Probleme konntet ihr feststellen?

- Für Wortvorgeschichten, die nicht im Datensatz auftreten, kann kein nächstes Wort vorgeschlagen werden.
- Die Wortvorschläge hängen nur vom vorherigen Wort ab und nicht vom erweiterten Satzkontext.

CAMMP Workshop | Wortvorschläge | 32

Ein erstes Sprachmodell

Welche Lösungsideen habt ihr?



- Für Wortvorgeschichten, die nicht im Datensatz auftreten, kann kein nächstes Wort vorgeschlagen werden.
- Die Wortvorschläge hängen nur vom vorherigen Wort ab und nicht vom erweiterten Satzkontext.

CAMMP Workshop | Wortvorschläge | 33

Ein erstes Sprachmodell

Welche Lösungsideen habt ihr?

- Für Wortvorgeschichten, die nicht im Datensatz auftreten, kann kein nächstes Wort vorgeschlagen werden.
 - Nicht auf die Wortvorgeschichte achten und das global häufigste Wort vorschlagen.
 - Größerer Datensatz
- Die Wortvorschläge hängen nur vom vorherigen Wort ab und nicht vom erweiterten Satzkontext.
 - Die Wortvorgeschichte um ein Wort (oder mehrere Wörter) erweitern.

CAMMP Workshop | Wortvorschläge | 34

Modellierungskreislauf

Reales Problem
Modell der Sprache, um Vorschläge zu geben
vereinfachen
Vereinfachtes Problem
Vereinfachtes Modell der Sprache, um Vorschläge zu geben
mathematisch beschreiben
Mathematisches Modell
berechnen
Mathematische Lösung
interpretieren

CAMMP Workshop | Wortvorschläge | 35

Jetzt seid ihr dran ...



Keine Panik!

ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Frage die Betreuenden!

CAMMP Workshop | Wortvorschläge | 36



Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Diskussion nach AB 2

Modellierungskreislauf

Reales Problem
Modell der Sprache, um Vorschläge zu geben
vereinfachen
Vereinfachtes Problem
Vereinfachtes Modell der Sprache, um Vorschläge zu geben
mathematisch beschreiben
Mathematisches Modell
berechnen
Mathematische Lösung
Übergangs-wahrscheinlichkeiten
interpretieren

Uni-Gramm-Modell: Wort 0 Wort 1
Bi-Gramm-Modell: Wort 0 Wort 1
Tri-Gramm-Modell: Wort 0 Wort 1

Wortvorschlag

Mathematisches Modell
Auswählen eines Wortes als Zufallsexperiment

CAMMP Workshop | Wortvorschläge | 38

A Lernmaterial zum Thema Wortvorhersage

Vergleich der *N*-Gramm-Modelle

Modell	Vorteile	Nachteile
Uni-Gramm-Modell		
Bi-Gramm-Modell		
Tri-Gramm-Modell		

 CAMMP Workshop | Wortvorschläge | 39

Vergleich der *N*-Gramm-Modelle

Modell	Vorteile	Nachteile
Uni-Gramm-Modell	immer ein Vorschlag, wenig Rechenaufwand	Vorschläge nicht kontextbasiert
Bi-Gramm-Modell		
Tri-Gramm-Modell		

 CAMMP Workshop | Wortvorschläge | 40

Vergleich der *N*-Gramm-Modelle

Modell	Vorteile	Nachteile
Uni-Gramm-Modell	immer ein Vorschlag, wenig Rechenaufwand	Vorschläge nicht kontextbasiert
Bi-Gramm-Modell	Vorschläge beziehen den Kontext nur wenig mit ein	geringer Kontextbezug, manchmal kein Vorschlag durch ungesehene Wortvorgeschichten
Tri-Gramm-Modell		

 CAMMP Workshop | Wortvorschläge | 41

Vergleich der *N*-Gramm-Modelle

Modell	Vorteile	Nachteile
Uni-Gramm-Modell	immer ein Vorschlag, wenig Rechenaufwand	Vorschläge nicht kontextbasiert
Bi-Gramm-Modell	Vorschläge beziehen den Kontext nur wenig mit ein	geringer Kontextbezug, manchmal kein Vorschlag durch ungesehene Wortvorgeschichten
Tri-Gramm-Modell	Vorschläge beziehen den Kontext mit ein	häufig kein Vorschlag durch ungesehene Wortvorgeschichten, hoher Rechenaufwand

 CAMMP Workshop | Wortvorschläge | 42

Vergleich der *N*-Gramm-Modelle

- Je **länger** die Wortvorgeschichte, desto besser passt der Wortvorschlag zum **Satzkontext**.
- Je **kürzer** die Wortvorgeschichte, desto weniger ungesehene Wortvorgeschichten gibt es und desto **verlässlicher** ist die Näherung der Übergangswahrscheinlichkeiten mit den relativen Häufigkeiten aufgrund höherer Auftretenshäufigkeiten.

 CAMMP Workshop | Wortvorschläge | 43

Vergleich der *N*-Gramm-Modelle

- Je **länger** die Wortvorgeschichte, desto besser passt der Wortvorschlag zum **Satzkontext**.
- Je **kürzer** die Wortvorgeschichte, desto weniger ungesehene Wortvorgeschichten gibt es und desto **verlässlicher** ist die Näherung der Übergangswahrscheinlichkeiten mit den relativen Häufigkeiten aufgrund höherer Auftretenshäufigkeiten.

➤ Wie können diese Erkenntnis nutzen, um unser Sprachmodell zu verbessern?



 CAMMP Workshop | Wortvorschläge | 44

Vergleich der *N*-Gramm-Modelle

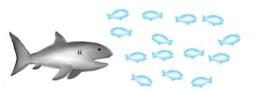
- Je **länger** die Wortvorgeschichte, desto besser passt der Wortvorschlag zum **Satzkontext**.
- Je **kürzer** die Wortvorgeschichte, desto weniger ungesehene Wortvorgeschichten gibt es und desto **verlässlicher** ist die Näherung der Übergangswahrscheinlichkeiten mit den relativen Häufigkeiten aufgrund höherer Auftretenshäufigkeiten.

➤ Alle drei Modelle zu einem Sprachmodell verknüpfen, um alle Vorteile zu nutzen:

- Zusatz: Rückfallstrategie
- Verknüpfung des Uni-, Bi- und Tri-Gramm-Modells über einen gewichteten Mittelwert

 CAMMP Workshop | Wortvorschläge | 45

Jetzt seid ihr dran ...



Keine Panik!



ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuenden!

 CAMMP Workshop | Wortvorschläge | 46



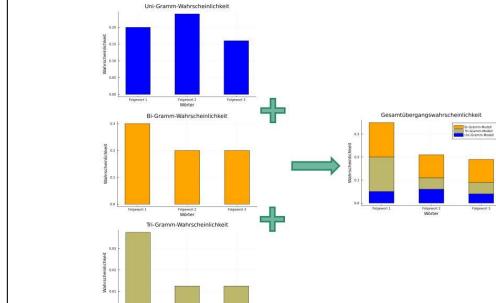
Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Diskussion nach AB 3

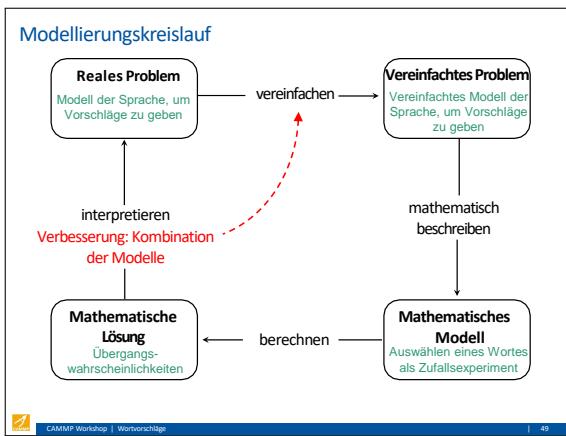


Verknüpfung der Modelle

Gewichteter Mittelwert der Uni-, Bi-, und Tri-Gramm-Wahrscheinlichkeit



| 48



CAMMP Workshop | Wortvorschläge

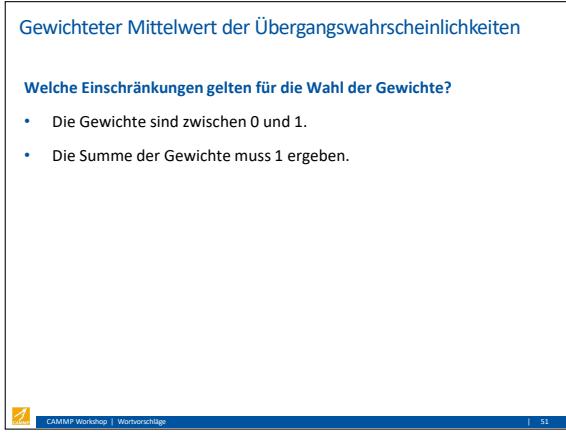
Gewichteter Mittelwert der Übergangswahrscheinlichkeiten

Welche Einschränkungen gelten für die Wahl der Gewichte?



CAMMP Workshop | Wortvorschläge

| 50



CAMMP Workshop | Wortvorschläge

Gewichteter Mittelwert der Übergangswahrscheinlichkeiten

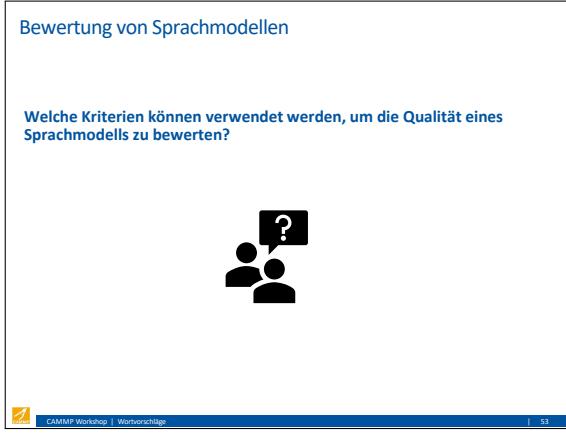
Welche Einschränkungen gelten für die Wahl der Gewichte?

- Die Gewichte sind zwischen 0 und 1.
- Die Summe der Gewichte muss 1 ergeben.

- Ziel: Gewichte so wählen, dass das Sprachmodell möglichst „gut“ ist.
➤ Wann ist ein Sprachmodell „gut“?

CAMMP Workshop | Wortvorschläge

| 52



CAMMP Workshop | Wortvorschläge

Bewertung von Sprachmodellen

Welche Kriterien können verwendet werden, um die Qualität eines Sprachmodells zu bewerten?

- Zeitersparnis
- Einsparung der Tastenanschläge
- Trefferrate
- Kreuzentropie

CAMMP Workshop | Wortvorschläge

| 54

Bewertung von Sprachmodellen

Für die meisten Qualitätsmaße werden sogenannte Testdatensätze benötigt.

```

graph TD
    A[Datengrundlage] --> B[Trainingsdatensatz]
    A --> C[Testdatensatz]
    B --> D[1. Teilung des Datensatzes]
    D --> E[2. Ermitteln der relativen Häufigkeiten als Parameter eines Modells]
    D --> F[3. Bewertung des Modells]
    C --> F
  
```

55

Die Kreuzentropie

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n))$$

- Modellwahrscheinlichkeit:** $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n)$ steht für die Wahrscheinlichkeit mit der aus unserem Modell ein Text bestehend aus den Wörtern Wort 1, Wort 2, Wort 3, Wort 4, ... entsteht.

56

Die Modellwahrscheinlichkeit

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

Hey, hast du Lust, morgen mit mir ins Kino zu gehen?

57

Die Kreuzentropie

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n))$$

- Modellwahrscheinlichkeit:** $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n)$ steht für die Wahrscheinlichkeit, mit der aus unserem Modell ein Text bestehend aus den Wörtern Wort 1, Wort 2, Wort 3, Wort 4, ... entsteht. große Modellwahrscheinlichkeit → Sprachmodell passt gut

58

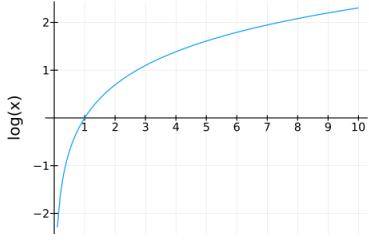
Die Kreuzentropie

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n))$$

- Modellwahrscheinlichkeit:** $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n)$ steht für die Wahrscheinlichkeit, mit der aus unserem Modell ein Text bestehend aus den Wörtern Wort 1, Wort 2, Wort 3, Wort 4, ... entsteht. große Modellwahrscheinlichkeit → Sprachmodell passt gut
- Logarithmusfunktion:** \log

59

Die Logarithmusfunktion



Welcher Definitionsbereich ist für uns interessant? Welche Werte nimmt die Funktion für diesen Definitionsbereich an?

60

Die Kreuzentropie

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n))$$

- Modellwahrscheinlichkeit:** $\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n)$ steht für die Wahrscheinlichkeit, mit der aus unserem Modell ein Text bestehend aus den Wörtern Wort 1, Wort 2, Wort 3, Wort 4, ... entsteht. große Modellwahrscheinlichkeit → Sprachmodell passt gut
- Logarithmusfunktion:** \log
- Anzahl der Wörter im Testdatensatz:** n

61

Die Kreuzentropie

$$K = -\frac{1}{n} \cdot \log(\mathbb{P}(\text{Wort 1}, \text{Wort 2}, \text{Wort 3}, \dots, \text{Wort } n))$$

➤ **Merke:**

Eine kleine Kreuzentropie bedeutet, dass das Sprachmodell gut zu unserem Testdatensatz passt. Wir schlussfolgern daraus, dass das Sprachmodell für diese Art von Texten den Sprachgebrauch gut modelliert.

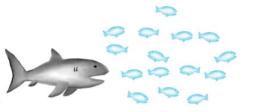
62

Vorteile der Kreuzentropie

- weniger Rechenaufwand als bei anderen Qualitätsmaßen
- Qualitätsmaß ist nicht nur für die Bewertung von Sprachmodellen, die zur Wortvorhersage genutzt werden, sondern auch für die Bewertung von Sprachmodellen unabhängig von deren Anwendungsbereich geeignet

 CAMMP Workshop | Wortvorschläge | 63

Jetzt seid ihr dran ...


Keine Panik!


ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuenden!

 CAMMP Workshop | Wortvorschläge | 65


Quelle: https://www.cammp.de/ai-in-our-life/

Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Diskussion nach AB 4

Glättung

Was ist die Glättung, und weshalb trägt sie zur Verbesserung des Sprachmodells bei?



 CAMMP Workshop | Wortvorschläge | 67

Glättung

Was ist die Glättung, und weshalb trägt sie zur Verbesserung des Sprachmodells bei?

Ungesehene Wörter erhalten ohne Glättung die Wahrscheinlichkeit null. Jedes Wort tritt aber mit einer gewissen, wenn vielleicht auch kleinen Wahrscheinlichkeit auf!

- Hinzufügen einer kleinen Wahrscheinlichkeit, die für jedes Wort im Vokabular gleich groß ist: $P_G = \frac{1}{V}$
- Geglättete Gesamtwahrscheinlichkeit:

$$P_{\text{ges}}(\text{Wort}) = g_0 \cdot \frac{1}{V} + (g_1 - \varepsilon) \cdot P_{\text{Uni}}(\text{Wort}) + (g_2 - \varepsilon) \cdot P_{\text{Bi}}(\text{Wort}) + (g_3 - \varepsilon) \cdot P_{\text{Tri}}(\text{Wort})$$

 CAMMP Workshop | Wortvorschläge | 68

Die Modellwahrscheinlichkeit

... ist ziemlich klein! Woran liegt das?

Inwiefern hilft das Logarithmieren der Modellwahrscheinlichkeit?



 CAMMP Workshop | Wortvorschläge | 69

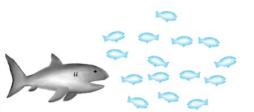
Vergleich der Modelle



- Wie groß ist die Kreuzentropie für die einzelnen Modelle, und was sagt dies über die Modelle im Vergleich aus?
- Wie groß ist die Kreuzentropie für die Kombination der Modelle, und was sagt dies über die Qualität im Vergleich mit den einzelnen Modellen aus?
- Welche Tendenz für eine sinnvolle Wahl der Gewichte hat sich gezeigt? Wie können wir die optimalen Gewichte bestimmen?

 CAMMP Workshop | Wortvorschläge | 70

Jetzt seid ihr dran ...


Keine Panik!


ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuenden!

 CAMMP Workshop | Wortvorschläge | 71



Künstliche Intelligenz im Alltag: Wortvorschläge am Handy

Abschlussdiskussion



Quelle: joostvan.com/photoblog/AnNaBBW

Bestimmung der optimalen Gewichte

- Welche Werte der Gewichte minimieren die Kreuzentropie für den vorliegenden Datensatz?
- Passt das Ergebnis zu den bisherigen Überlegungen?

Bestimmung der optimalen Gewichte

- Annahme: Datensatz repräsentiert Alltagswortschatz gut → die ermittelten Gewichte sind im Allgemeinen sinnvoll
- Vorsicht: Trainingsdatensatz Testdatensatz

Grenzen des Sprachmodells

Heute morgen war es total sonnig, und dann hat es trotzdem noch angefangen zu ____.

Jugendwörter der letzten Jahre:
„slay“, „cringe“, „goofy“

Grammatikalische Regeln

Grenzen des Sprachmodells

Kritische Diskussion

Welche Aspekte sollten bei der Entwicklung und Nutzung von Sprachmodellen kritisch hinterfragt werden?



- Beeinflussen Wortvorschläge das Schreibverhalten? In welcher Art? Ist die Einflussnahme den Nutzenden bewusst?
- Wie viel Energieverbrauch darf für innovative Forschung an Sprachmodellen in Kauf genommen werden?
- Können Sprachmodelle und deren Anwendung in Chatbots und Assistenzsystemen Vorurteile verstärken und diskriminierend sein? Wie kann dagegen vorgegangen werden? Sollte dagegen vorgegangen werden?

CAMMP Workshop | Wortvorschläge | 76

Bestimmung der optimalen Gewichte

- Welche Werte der Gewichte minimieren die Kreuzentropie für den vorliegenden Datensatz?
- Passt das Ergebnis zu den bisherigen Überlegungen?

Grenzen des Sprachmodells

Kritische Diskussion

Welche Aspekte sollten bei der Entwicklung und Nutzung von Sprachmodellen kritisch hinterfragt werden?

Kritische Diskussion: Beeinflussung durch Assistenzsysteme

Studie zum Einfluss von Assistenzsystemen bei der Texteingabe am Beispiel von Restaurant-Kritiken zeigt:

Personen, die mit positiv verzerrten Vorschlägen für die Texteingabe konfrontiert wurden, schreiben mehr positive Bewertungen als Personen, die negativ-verzerrte Vorschläge bekommen.¹

¹ K. C. Arnold, K. Chauncey, and K. Gajos. Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. Proceedings of Graphics Interface 2018, S. 42 - 49, 2018. doi: 10.20380/GI2018.07

CAMMP Workshop | Wortvorschläge | 79

288

Kritische Diskussion: Ressourcenverbrauch

GPT-3, eines der bekanntesten und fortschrittlichsten Sprachmodelle, benötigt während eines einzigen Trainings schätzungsweise 1.287 Megawattstunden.² Das ist so viel Strom wie etwa 312 österreichische Haushalte im Jahr verbrauchen.



²Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., ... & Dean, J. (2021). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

 CAMMP Workshop | Wörtervorschläge | 80

Kritische Diskussion: Diskriminierung

Der selbstlernende Chat Bot Tay von Microsoft wurde durch das Lernen mit Texten von Twitter Nutzern und Nutzerinnen mit diskriminierenden Inhalten konfrontiert und erzeugte dadurch rassistische, antisemitische und sexistische Antworten.³



³K. C. Arnold, K. Chauncey, and K. Gajos. Sentiment Bias in Predictive Text Recommendations Results in Biased Writing. *Proceedings of Graphics Interface 2018*, S. 42 - 49, 2018. doi: 10.20380/GI2018.07

 CAMMP Workshop | Wörtervorschläge | 81

Ziel und Ablauf

Ablauf:

1. Ein Sprachmodell auf einem kleinen Trainingsdatensatz entwickeln.
2. Die Erstellung des Sprachmodells automatisieren. Das Sprachmodell auf einem größeren Datensatz anwenden und stetig verbessern!
3. Das entwickelte Sprachmodell bewerten.
4. Kritisch reflektieren: Was sollte bei der Entwicklung und Nutzung von Sprachmodellen kritisch reflektiert werden?



 CAMMP Workshop | Wörtervorschläge | 82

Verabschiedung

 Danke für die 

Teilnahme	Aufmerksamkeit
-----------	----------------

 CAMMP Workshop | Wörtervorschläge | 83

B Lernmaterial zum Thema Spracherkennung

Auf den nachfolgenden Seiten sind die digitalen Arbeitsblätter mit eingetragenen Lösungen abgedruckt. Alle ausklappbaren Lösungen und Tipps sind hier ausgeklappt. Für eine Exploration der interaktiven Elemente sei auf die Lernplattform¹ verwiesen, auf der die digitalen Arbeitsblätter mit allen zugehörigen Zusatzblättern und Tippkarten in ihrer vollen Funktion eingesehen werden können. Auch für die Einsicht aller weiteren Begleitmaterialien sei an dieser Stelle auf die Lernplattform verwiesen.

Die Notation in den Codefeldern der Arbeitsblätter ist wie folgt zu verstehen:

- Der Platzhalter `#= None`; `=#`, der auch in abgewandelter Form erscheinen kann, kennzeichnet die Stellen, an denen die Lernenden Eingaben ergänzen müssen.
- Die Eingaben, die nach diesem Platzhalter folgen, stellen Beispiellösungen dar. Sie bieten Lehrkräften, die das Material im Unterricht oder in Projekten durchführen wollen, eine Kontrollmöglichkeit.

¹siehe workshops.cammp.online

B.1 Arbeitsblatt 1

Arbeitsblatt 1 | Einen Ton mathematisch modellieren



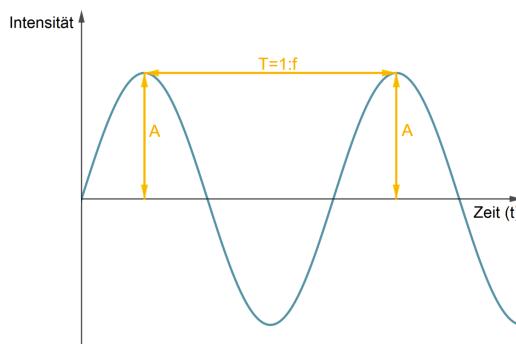
In diesem Workshop werden wir ein Modell entwickeln, mit dem verschiedene Wörter anhand bestimmter Merkmale ihres Sprachsignals erkannt werden können. Hierfür betrachten wir zunächst was Sprachsignale sind, wie sie vom Menschen erzeugt und schließlich mathematisch beschrieben werden können.

Geräusche, wie Musik oder auch Sprache, sind Schallwellen. Eine Schallwelle ist eine Schwankung des Drucks in der Luft, welche sich wellenförmig ausbreitet. Das wohl einfachste Geräusch ist der (reine) Ton, dessen erzeugende Schwingung durch eine **Sinusfunktion** beschrieben werden kann. Die von der menschlichen Stimme erzeugten Geräusche setzen sich aus vielen verschiedenen Tönen zusammen. Diese Töne werden zum Beispiel durch das Schwingen der Stimmbänder erzeugt.

Auf diesem Aufgabenblatt wollen wir uns genauer damit beschäftigen, wie reine Töne mathematisch modelliert werden: Ein Ton g ist eine Sinusschwingung, deren **Amplitude** die Lautstärke des Tons festlegt und deren **Frequenz** der Tonhöhe entspricht. Er kann durch die Gleichung

$$g(t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t)$$

beschrieben werden. Hier sieht ihr den Graphen einer Sinusschwingung:



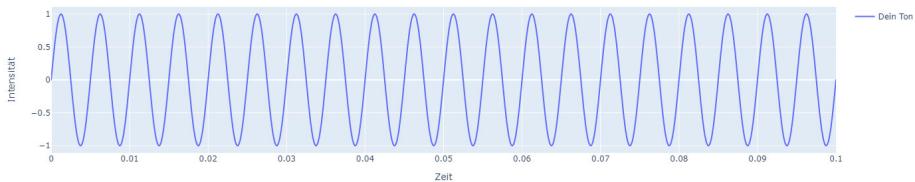
Dabei steht

- t für die **Zeit**
- A für die **Amplitude**. Diese gibt die maximale Auslenkung der Schwingung aus der Nulllage an. Physikalisch entspricht sie der **Lautstärke** des Tons. Je größer die Amplitude, desto lauter der Ton.
- f für die **Frequenz**. Die Frequenz gibt die Anzahl der Schwingungen pro Sekunde an. Die Einheit der Frequenz ist Hertz [Hz = 1/s]. Sie ist der Kehrwert der **Periodendauer** T ($T = 1 : f$). Die Periodendauer bezeichnet den zeitlichen Abstand zweier aufeinander folgender Maxima (oder auch zweier aufeinander folgender Minima) und besitzt folglich die Einheit Sekunde [s]. Dies bedeutet, je größer die Frequenz f , desto kleiner die Periodendauer - also der Abstand zwischen den Maxima. Physikalisch entspricht die Frequenz der **Tonhöhe**.

```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern! Hier werden alle Funktionen für dieses Arbeitsblatt geladen.
import sys; sys.path.append('../code'); from setup_AB1 import *
```

1) Die Amplitude eines Tons

Schau dir den folgenden Graphen, der einen Sinuston auf dem Intervall $[0, 0.1]$ zeigt, genau an:



- Welchen Wert hat die Amplitude dieses Tons?
- Welchen Wert hat die Frequenz dieses Tons?



a) Amplitude und Frequenz am Graphen ablesen

Ersetze im folgenden Codefeld `None` durch die Amplitude (engl. amplitude) bzw. die Frequenz (engl. frequency) des obigen Tons. Führe danach das Codefeld aus indem du auf den "Run"-Button klickst (oder Shift + Enter drückst).

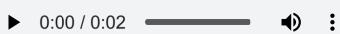


Daraufhin kannst du den oben abgebildeten Sinuston anhören. Außerdem wird der Graph des Sinustons gezeigt, der die von dir eingegebene Amplitude und Frequenz besitzt. Vergleiche die beiden Graphen. Beachte dabei die Achsenkalierung. Falls die Töne nicht identisch sind, bekommst du eine Fehlermeldung. Du kannst deine Eingabe jederzeit überschreiben und das Codefeld nochmals ausführen. Nutze den Graphen um dich gegebenenfalls zu korrigieren.

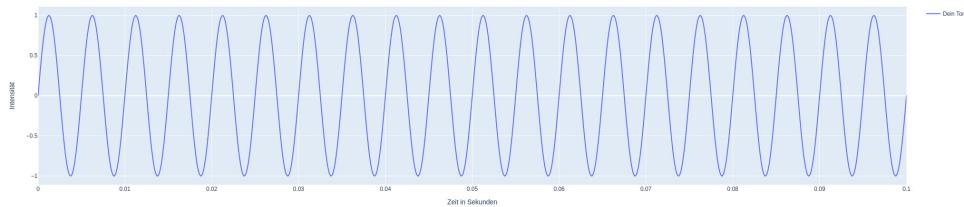
! Achte darauf, dass bei der Aufgabe die Lautstärke am Laptop nicht höher als bis zur Hälfte aufgedreht ist!

Tipp 1 Tipp 2

```
In [2]: # Gib die Amplitude und Frequenz des oben angezeigten Tons an
# Ersetze 'None' durch die Amplitude des oben abgebildeten Tons
amplitude = 1 #=None#
# Ersetze 'None' durch die Frequenz des oben abgebildeten Tons
frequency = 200 #=None#
# Ab hier nichts mehr ändern
play_given_tone()
plot_your_tone(amplitude, frequency)
```



✓ Super! Amplitude und Frequenz stimmen.



Wir wissen bereits: Je größer die **Amplitude** gewählt wird, desto lauter wird der Ton wahrgenommen. Dies nutzen wir nun um einen lauteren bzw. leiseren Ton als in Aufgabenteil a) zu erzeugen.



b) Lauter und leiser Ton

Ersetze im folgenden Codefeld `None` durch geeignete Amplituden, sodass der Ton mit Amplitude `amplitude_quiet_tone` leiser ist als der Ton aus Aufgabenteil a und der Ton mit Amplitude `amplitude_loud_tone` lauter ist als der Ton aus Aufgabenteil a). Führe das Codefeld aus. Als Ausgabe erhältst du zuerst das Audiosignal des Tons aus Teil a, dann des Tons mit `amplitude_quiet_tone` und dann des Tons mit `amplitude_loud_tone`. Die Töne werden auch graphisch dargestellt: Der Ton aus Aufgabenteil a) in Blau, der Ton mit Amplitude `amplitude_quiet_tone` in Rot und der mit `amplitude_loud_tone` in Grün. Korrigiere deine Eingabe bei einer Fehlermeldung mit Hilfe des Graphen und des Audiosignals.

Wenn du Dezimalzahlen verwenden möchtest, musst du anstelle des Kommas einen Punkt schreiben: Statt 1,7 schreibe 1.7

! Diese Aufgabe besitzt mehr als eine richtige Lösung. Probiere gerne verschiedene Lösungen aus und siehe/höre dir die Ergebnisse an.

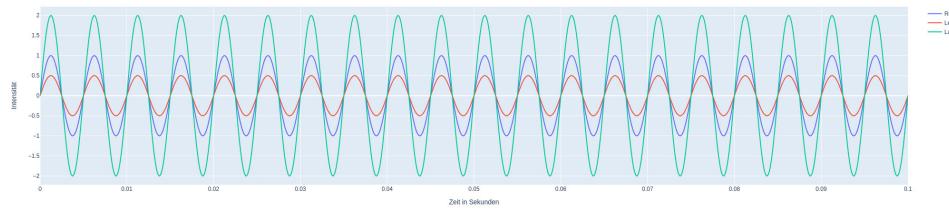
Tipp 1 Tipp 2

In [3]: `# Lautere und Leisere Töne`

```
# Ersetze 'None' durch einen Wert, sodass der Ton im Vergleich zu Teil a Leiser wird
amplitude_quiet_tone = 0.5 #=None=#
# Ersetze 'None' durch einen Wert, sodass der Ton im Vergleich zu Teil a lauter wird
amplitude_loud_tone = 2 #=None=#
# Ab hier nichts mehr ändern
play_three_tones(amplitude_quiet_tone, amplitude_loud_tone)
plot_three_tones(amplitude, amplitude_quiet_tone, amplitude_loud_tone)
```

▶ 0:00 / 0:08 ━━━━ 🔊 ⏮

✓ Super! Du hast die Amplituden entsprechend gewählt!



Bisher war die Amplitude des Tons über die Zeit konstant. Nun soll die Lautstärke des Tons mit der Zeit variieren. Der Ton besitzt also die Form

$$g(t) = A(t) \cdot \sin(2 \cdot \pi \cdot f \cdot t),$$

mit einer von der Zeit abhängigen Amplitude A .



c1) Lautstärke variieren

Bestimme die Amplitude in Abhängigkeit der Zeit t , sodass die Lautstärke des zugehörigen Tons zum Zeitpunkt $t = 0$ den Wert Null hat und lineares Wachstum in t besitzt. Ersetze im folgenden Codefeld `None` durch diesen Ausdruck für die Amplitude in Abhängigkeit von t . Führe das Codefeld aus, um dir deinen erzeugten Ton anzuhören und anzusehen.

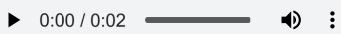


Du kennst bereits viele Funktionen mit linearem Wachstum, nämlich alle ansteigenden Geraden.

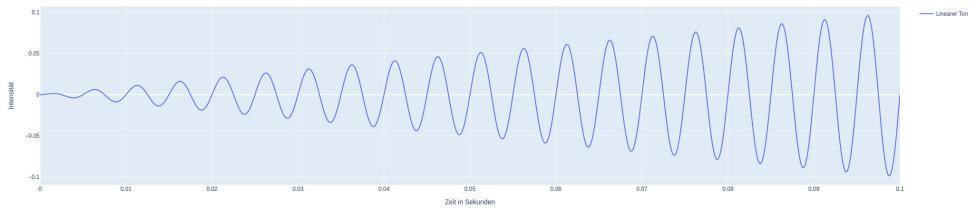


Tipp 2

```
In [4]: # Lineares Wachstum der Lautstärke
def linear_amplitude(t):
    # Ersetze 'None' durch einen Ausdruck in t, der lineares Wachstum in t beschreibt und bei t = 0 den
    # return t #= return None=#
    # Ab hier nichts mehr ändern
    play_linear_tone(linear_amplitude)
    plot_linear_tone(linear_amplitude)
```



✓ Super! Deine Eingabe ist korrekt.



c2) Lautstärke variieren

Überlege dir, wie die zeitabhängige Amplitude eines Tons gewählt werden muss, der folgende Bedingungen erfüllt:

- Seine Amplitude wächst linear.
- Bei $t = 0$ hat seine Amplitude den Wert 0.
- Seine Amplitude wächst schneller als die des Tons, den du in Aufgabenteil c1) erstellt hast.

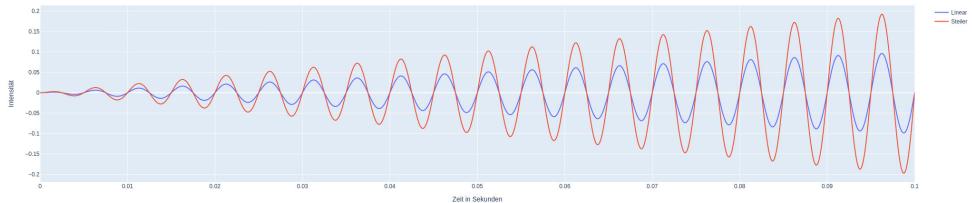
Die richtige Lösung hängt hier also davon ab, was du in der letzten Teilaufgabe eingegeben hast. Ersetze im folgenden Codefeld `None` durch einen Term in t , der die Amplitude eines Tons beschreibt, welcher die obigen drei Bedingungen erfüllt. Führe das Codefeld aus. Es werden hintereinander beide Töne abgespielt: Erst der Ton aus Aufgabenteil c1) und dann der aus Aufgabenteil c2).



```
In [5]: # Steilerer Wachstum der Lautstärke
def steeper_amplitude(t):
    # Ersetze 'None' durch einen Term in t, der die Amplitude eines Tons beschreibt, welcher die obigen
    # Bedingungen erfüllt.
    return 2 * t #= return None=#
# Ab hier nichts mehr ändern
play_two_tones(linear_amplitude, steeper_amplitude)
plot_two_tones(linear_amplitude, steeper_amplitude)
```

▶ 0:00 / 0:05 ━━━━ 🔊 ⏮

✓ Super! Deine Eingabe ist korrekt.



d) Linear fallende Amplitude

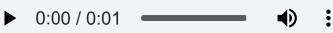
Erzeuge nun einen Ton, dessen Amplitude linear fällt und dessen Länge 1 s beträgt. Ersetze dazu im folgenden Codefeld `None` durch einen passenden Ausdruck für die Amplitude in Abhängigkeit von t . Führe das Codefeld aus, um dir deinen erzeugten Ton anzu hören und anzusehen.



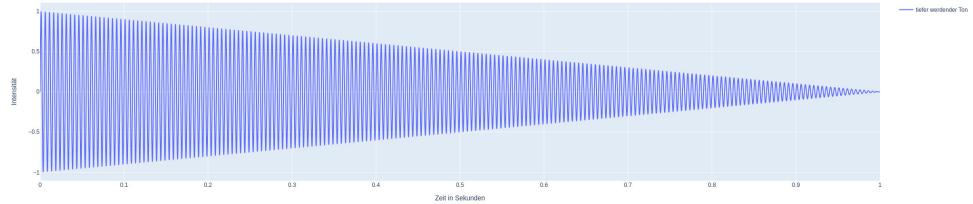
Überlege dir dazu, wie die Amplitude eines Tons in Abhängigkeit der Zeit t gewählt werden muss, damit diese linear fällt und bei $t = 1$ den Wert 0 hat.



```
In [6]: # Linear abfallende Lautstärke
def decreasing_amplitude(t):
    # Ersetze 'None' durch einen Term in t, der die Amplitude eines Tons beschreibt, der 1 s lang ist und
    return 1 - t # = return None # Ab hier nichts mehr ändern
play_decreasing_tone(decreasing_amplitude)
plot_decreasing_tone(decreasing_amplitude)
```



✓ Super! Deine Eingabe ist korrekt.



2) Die Frequenz eines Tons

Bisher hatten alle betrachteten Töne eine Frequenz von 200 Hz. Nun werden wir dazu übergehen, nicht nur die Amplitude, sondern auch die Frequenz eines Tons zu ändern. Du weißt bereits: Je höher ein Ton klingt, desto höher ist seine **Frequenz** und desto kleiner ist seine Periodendauer. Betrachte den Ton

$$g(t) = \sin(F \cdot 200 \cdot 2\pi \cdot t), \quad t \geq 0,$$

dessen Frequenz vom Faktor F abhängt. Betrachte außerdem den Referenzton

$$\sin(200 \cdot 2\pi \cdot t), t \geq 0.$$



a) Hoher und tiefer Ton

Wähle den Faktor F so, dass er einmal höher und einmal tiefer als der Referenzton klingt. F soll dabei in beiden Fällen - das heißt, für den höheren und den tieferen Ton - zwischen 0 und 6 liegen. Ersetze dazu im folgenden Codefeld die zwei `None` durch einen Wert zwischen 0 und 6, sodass der Ton g , mit dem von dir gewählten $F = f_{\text{higher}}$, höher klingt als der Referenzton und der Ton, mit dem von dir gewählten $F = f_{\text{lower}}$, tiefer

klingt als der Referenzton. Führe das Codefeld aus, um die drei Töne abzuspielen: erst den Referenzton, dann den Ton mit `f_higher`, zum Schluss den Ton mit `f_lower`.

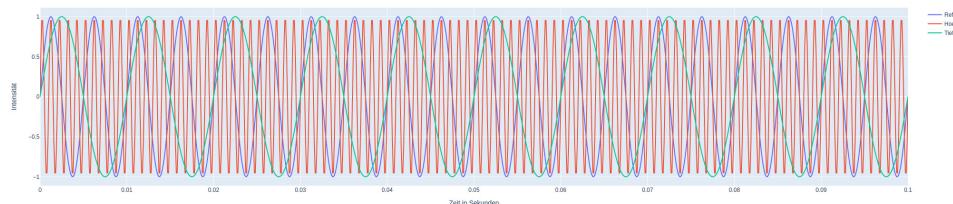
! Diese Aufgabe besitzt mehr als eine richtige Lösung. Probiere gerne verschiedene Lösungen aus und siehe/höre dir die Ergebnisse an.

In [7]: `# Höhere und tiefere Töne`

```
# Ersetze 'None' durch einen geeigneten Wert für F, sodass der Ton g (s.o.) höher klingt, als der Referenzton
f_higher = 5 #=None#
# Ersetze 'None' durch einen geeigneten Wert für F, sodass der Ton g (s.o.) tiefer klingt, als der Referenzton
f_lower = 0.5 #=None#
play_high_low_tones(f_higher, f_lower)
plot_high_low_tones(f_higher, f_lower)
```

▶ 0:00 / 0:08 🔍 ⏮

✓ Super! Deine Eingabe ist korrekt.



Wir betrachten nun den Ton

$$g(t) = \sin(F(t) \cdot 200 \cdot 2\pi \cdot t), t \geq 0.$$

Diesmal ist die Frequenz F jedoch keine Konstante, sondern ebenfalls von der Zeit t abhängig.



b) Linear wachsende Frequenz

Bestimme $F(t)$ so, dass

- die Frequenz des akustischen Signals g linear wächst,
- das Signal g zum Zeitpunkt $t = 0$ eine Frequenz von 200 Hz besitzt
- das Signal g zum Zeitpunkt $t = 0.1$ eine Frequenz von 400 Hz besitzt.

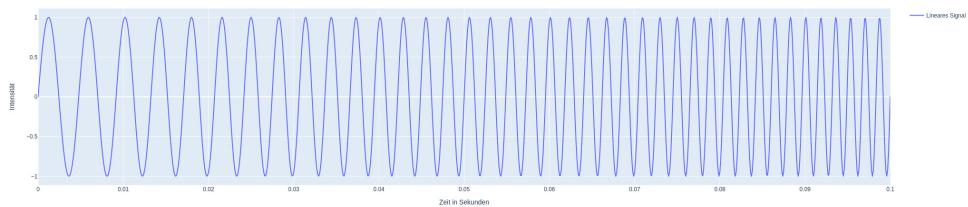
Ersetze im folgenden Codefeld 'None', durch einen passenden Ausdruck für $F(t)$. Führe das Codefeld aus, um dir den Ton anzuhören und anzusehen.

! Da sich die Frequenz von g mit der Zeit ändert, handelt es sich bei g nicht mehr um einen reinen Ton!

Tipp 1 Tipp 2

```
In [8]: # Linear wachsende Frequenz
def F(t):
    # Ersetze 'None' durch einen Term in t, sodass der Faktor F linear wächst und der Ton g bei t = 0 (l
    return 10 * t + 1 #= return None=#
    # Ab hier nichts mehr ändern
play_signal(F)
plot_signal(F)
```

✓ Super! Deine Eingabe ist korrekt.



Fazit

Du hast auf diesem Arbeitsblatt kennengelernt, wie Töne mathematisch durch Sinusfunktionen beschrieben werden können. Auf dem nächsten Arbeitsblatt werden wir erkunden, wie charakteristische Merkmale aus einem Ton extrahiert werden können.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 1 gemeinsam im Plenum besprechen.

Öffne nun das nächste Arbeitsblatt.

Schon fertig?

Schau dir auf diesem [Zusatzblatt](#) komplexere Signale und ihre mathematische Darstellung an.



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Autorinnen und Autoren: Maike Gerhard und Lars Schmidt

B.2 Arbeitsblatt 2

Arbeitsblatt 2 | Die Fourier-Transformation am Beispiel von Dreiklängen



Um gesprochene Wörter zu erkennen, werden bei Alexa, Siri und co. zunächst charakteristische Merkmale aus den Sprachsignalen extrahiert. Charakteristisch für ein Sprachsignal sind vor allem die enthaltenen Frequenzen. Diese werden Zeitschritt für Zeitschritt aus dem Signal ermittelt. Allerdings ist ein Sprachsignal sehr viel komplexer als ein Sinuston, sodass die Bestimmung der enthaltenen Frequenzen nicht durch einfaches Ablesen der Periodendauer geschehen kann.

Auf diesem Arbeitsblatt werden wir daher eine Theorie bzw. Methode kennenlernen, die es ermöglicht, sogar die kompliziertesten akustischen Signal in einzelne Sinustöne zu zerlegen: die **Fourier-Analyse** bzw. die **Fourier-Transformation**. Sie ist nach dem französischen Mathematiker Joseph Fourier benannt, dessen Werk die moderne Physik und Technik bis heute beeinflusst.



Abbildung 3: Jean Baptiste Joseph Fourier (1768 - 1830)
(https://de.wikipedia.org/wiki/Joseph_Fourier)

```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern!
import sys; sys.path.append('../code'); from setup_AB2 import *; from math import pi; from numpy import
```

Um die grundlegende Idee hinter der Fourier-Transformation zu verstehen, betrachten wir Dreiklänge. Unter einem Dreiklang versteht man in der Musik das gleichzeitige Erklingen dreier unterschiedlicher Töne mit einem bestimmten Intervallabstand. Obwohl diese akustischen Signale auf den ersten Blick unspektakulär erscheinen mögen - sind sie doch nicht viel komplexer als ein Sinuston -, bilden sie die Grundlage der westlichen Musik (Rock, Pop, Klassik, Schlager, etc.). Sie können mathematisch durch die Addition dreier geeigneter Sinusschwingungen modelliert werden.

Hier ist ein Beispiel für einen Dreiklang, der die Frequenzen 440 Hz, 277 Hz und 330 Hz enthält:

$$S(t) = \sin(440 \cdot 2\pi \cdot t) + \sin(277 \cdot 2\pi \cdot t) + \sin(330 \cdot 2\pi \cdot t). \quad (1)$$

Wir beschränken uns in diesem Workshop auf Dur- und Molldreiklänge. Bei einem Durdreiklang liegen zwischen dem Grundton (das ist der tiefste Ton in der Grundstellung eines Dreiklangs) und dem mittleren Ton vier Halbtöne; zwischen dem mittleren und dem oberen Ton liegen drei Halbtöne. Im Gegensatz dazu liegen beim Molldreiklang drei Halbtöne zwischen Grundton und mittlerem Ton und vier Halbtöne zwischen dem mittleren und dem oberen Ton. Tabelle 2 zeigt, welche Dreiklänge sich nach diesem Prinzip über den zwölf Grundtönen der chromatischen Tonleiter (s. Tabelle 1) bilden lassen. Die Frequenzen der einzelnen (eingestrichenen) Töne können in Tabelle 1 abgelesen werden. Die angegebenen Werte sind gerundet und setzen eine gleichstufige Stimmung voraus, das bedeutet, die Frequenzen zweier aufeinanderfolgender Töne unterscheiden sich um den Faktor $\sqrt[12]{2} \approx 1.059$.

Ton	Frequenz in Hz	Grundton	Durdreiklang Name	Akkord	Molldreiklang Name	Akkord
c	262	c	C-Dur	c-e-g	C-Moll	c-es-g
cis/des	277	cis/des	Des-Dur	des-f-as	Des-Moll	des-fes-as
d	294	d	D-Dur	d-fis-a	D-Moll	d-f-a
dis/es	311	dis/es	Es-Dur	es-g-b	Es-Moll	es-ges-b
e/fes	330	e/fes	E-Dur	e-gis-h	E-Moll	e-g-h
f	349	f	F-Dur	f-a-c	F-Moll	f-as-c
fis/ges	370	fis/ges	Ges-Dur	ges-b-des	Fis-Moll	fis-a-cis
g	392	g	G-Dur	g-h-d	G-Moll	g-b-d
gis/as	415	gis/as	As-Dur	as-c-es	As-Moll	as-ces-es
a	440	a	A-Dur	a-cis-e	A-Moll	a-c-e
ais/b	466	ais/b	B-Dur	b-d-f	B-Moll	b-des-f
h/ces	494	h/ces	H-Dur	h-dis-fis	H-Moll	h-d-fis

Tabelle 1 Frequenzen der eingestrichenen Töne (gleichstufige Stimmung)

Tabelle 2 Dur- und Molldreiklänge über verschiedenen Grundtönen (Zur besseren Lesbarkeit wurde auf die enharmonische Verwechslung verzichtet.)

Gleichung 1 beschreibt demzufolge einen A-Dur Dreiklang.

Ziel ist es, im Folgenden eigene Dreiklänge zu erstellen sowie die Frequenzen unbekannter Dreiklänge mithilfe der Fourier-Transformation zu bestimmen.

1) Dreiklänge erstellen

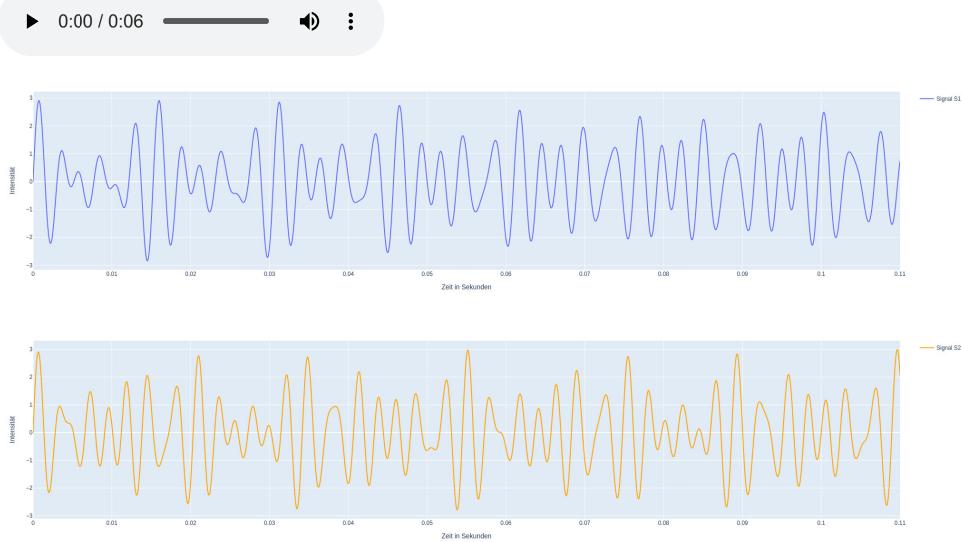
Du sollst nun selbst zwei beliebige Dreiklänge aus Tabelle 2 implementieren. Bilde dazu jeweils die Summe von drei Sinusfunktionen mit geeigneten Frequenzen.



Ersetze im folgenden Code die `None` durch zwei Dreiklänge aus Tabelle 2. Drücke anschließend auf "Run", damit deine Dreiklänge graphisch dargestellt und abgespielt werden - zuerst `S1`, dann `S2`.

 Du kannst das Zeichen π durch `pi` im Code eingeben. Da es viele richtige Lösungen gibt, wird die Eingabe nicht kontrolliert.

```
In [2]: def S1(t):
    # Ersetze 'None' durch deinen ersten Dreiklang. Benutze dazu Tabelle 2.
    return sin(262 * 2 * pi * t) + sin(330 * 2 * pi * t) + sin(392 * 2 * pi * t) #= return None=#
def S2(t):
    # Ersetze 'None' durch deinen zweiten Dreiklang. Benutze dazu Tabelle 2.
    return sin(294 * 2 * pi * t) + sin(349 * 2 * pi * t) + sin(440 * 2 * pi * t) #= return None=#
# Ab hier nichts mehr ändern
play_triads(S1, S2)
plot_triads(S1, S2)
```



2) Dreiklänge erkennen

In Aufgabe 1 hast du Sinustöne zu Dreiklängen zusammengesetzt. Nun wird die umgekehrte Problemstellung betrachtet: Du sollst einen unbekannten Dreiklang in seine elementaren Töne zerlegen.

Das ist gar nicht so einfach, ohne die richtigen mathematischen Werkzeuge. Am Graphen ablesen kann man die einzelnen Frequenzen jedenfalls nicht mehr so einfach. Oder etwa doch?



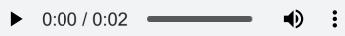
a) Frequenzen ablesen

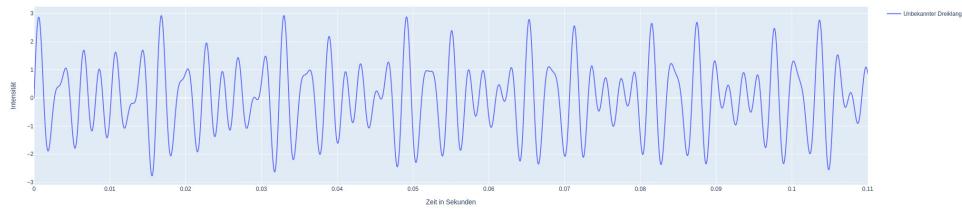
Führe den folgenden Code aus. Daraufhin kannst du dir einen Dreiklang anhören und den zugehörigen Graphen ansehen. Versuche nur anhand von Klang und Graph herauszufinden, welche drei Töne bzw. Frequenzen im Dreiklang enthalten sind. Notiere deine Vermutung im untenstehenden Freitextfeld.

! Die korrekte Lösung wird erst weiter unten verraten.

Tipp 1 Tipp 2

```
In [3]: # Hier nichts ändern, nur ausführen!
play_unknown_triad0(); plot_unknown_triad0(); prompt_1()
```





Deine abgelesenen Frequenzen:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

 Speichern

Du wirst festgestellt haben, dass die gerade angewandte Methode mühsam und fehleranfällig ist. Das gilt erst recht für noch kompliziertere akustische Signale. Wir müssen uns etwas anderes einfallen lassen.

Akustische Signale in Sinustöne zerlegen

Das passende mathematische Werkzeug ist in diesem Fall die Fourier-Transformation. Dabei wird die Tatsache ausgenutzt, dass man **beliebige akustische Signale - zumindest nährungsweise - als Summe von Sinusschwingungen verschiedener Frequenzen darstellen kann**. Die Fourier-Transformation "zerlegt" ein Audiosignal in seine einzelnen Sinustöne und bestimmt die Frequenzen und Amplituden dieser Töne.

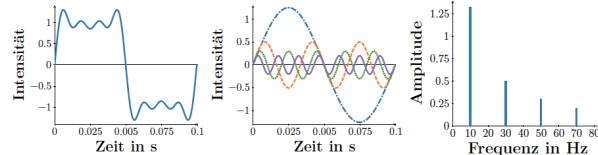


Abbildung 4: Zerlegung eines Signals in einzelne Frequenzen

Obige Abbildung verdeutlicht das Prinzip hinter der Fourier-Transformation: Das linke Bild zeigt das akustische Ausgangssignal. Das mittlere Bild zeigt die Zerlegung dieses Signals in einzelne Sinusschwingungen. Es ist gut zu erkennen, dass das akustische Signal die Summe von 4 Sinustönen (blau, rot, grün, violett) mit unterschiedlichen Frequenzen und Amplituden ist. Trägt man die Amplitude jedes Teiltons gegen seine Frequenz auf, erhält man den im Bild rechts abgebildeten Graphen. Einen solchen Graphen nennt man **Amplitudenspektrum**. An diesem kann leicht abgelesen werden, welche Frequenzen zu welchem Anteil in einem Signal enthalten sind. Insgesamt bewirkt die Fourier-Transformation also einen Wechsel vom Zeitraum (im Bild links) in den Frequenzraum (im Bild rechts).

Die **Fourier-Transformation** ist damit genau das Werkzeug, das wir zum Zerlegen und damit zur **Identifikation von Dreiklängen** brauchen. Da ein Dreiklang aus drei Tönen zusammengesetzt ist, enthält sein Amplitudenspektrum genau drei Peaks.

Nun sollst du den unbekannten Dreiklang aus Aufgabenteil a) noch einmal identifizieren - diesmal steht dir dabei jedoch die Fourier-Transformation zur Verfügung. Der folgende Code generiert zwei Abbildungen. Die erste zeigt den bereits bekannten Graphen des Dreiklangs. Die zweite zeigt dessen Amplitudenspektrum, das unter Verwendung der Fourier-Transformation berechnet wurde.



b) Frequenzen eines ersten Dreiklangs mit der Fourier-Transformation bestimmen

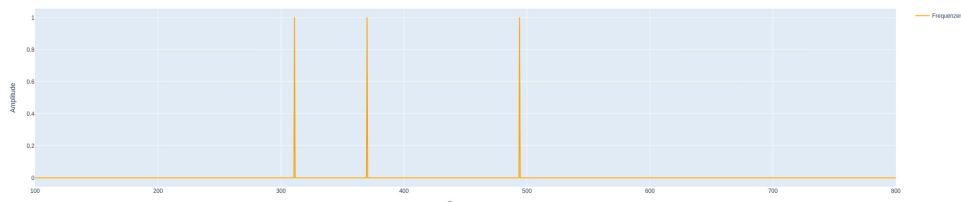
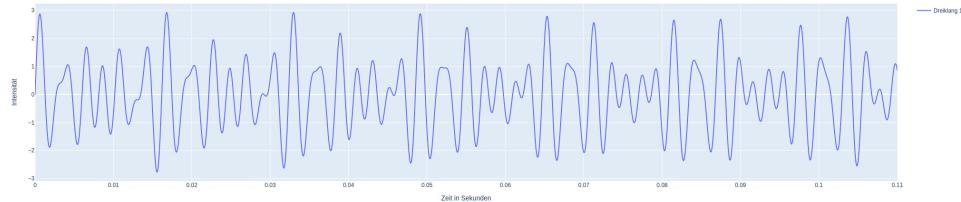
Führe den folgenden Code aus, um dir den gesuchten Dreiklang anzuhören und im Zeit- und Frequenzraum anzusehen. Bestimme die Frequenzen des Signals und wähle im Dropdownmenü unter der Grafik den zugehörigen Dreiklang aus. Vergleiche dein Ergebnis mit deinem Ergebnis aus Aufgabenteil a).

Hier findest du die Übersicht der Dreiklänge.

Wenn du mit dem Mauszeiger über die Graphen fährst, werden dir die Koordinaten (x-Koordinate, y-Koordinate) der gezeichneten Datenpaare angezeigt.

```
In [4]: # Hier nichts ändern, nur ausführen!
play_unknown_triad1()
plot_unknown_triad1()
check_unknown_triad1()
```

▶ 0:00 / 0:02 ⏸ ⏴ ⏵



Dreiklang: C-Dur

Überprüfen



c) Frequenzen eines zweiten Dreiklangs mit der Fourier-Transformation bestimmen

Bestimme einen weiteren Dreiklang mit Hilfe der Fourier-Transformation. Führe den folgenden Code aus, um dir den gesuchten Dreiklang anzuhören und im Zeit- und Frequenzraum anzusehen. Bestimme die Frequenzen des Signals und wähle im Dropdownmenü unter der Grafik den zugehörigen Dreiklang aus.

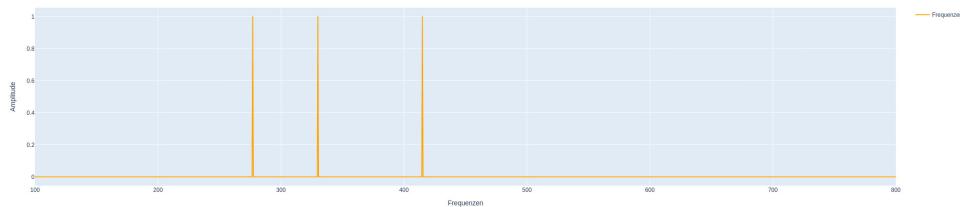
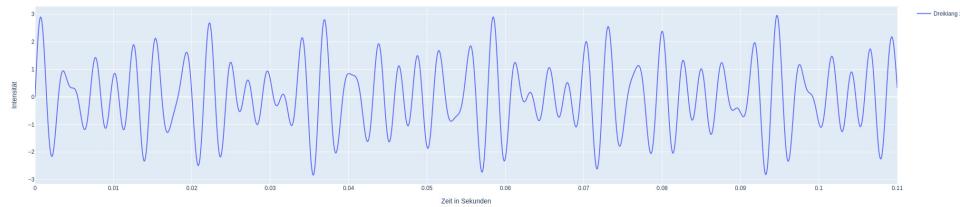


Hier findest du die Übersicht der Dreiklänge.

Wenn du mit dem Mauszeiger über die Graphen fährst, werden dir die Koordinaten (x-Koordinate, y-Koordinate) der gezeichneten Datenpaare angezeigt.

In [5]: *# Hier nichts ändern, nur ausführen!*
`play_unknown_triad2(); plot_unknown_triad2(); check_unknown_triad2()`

▶ 0:00 / 0:02 ⏪ ⏹ ⏷



Dreiklang: ▾

✓ Überprüfen



d) Frequenzen eines dritten Dreiklangs mit der Fourier-Transformation bestimmen

Bestimme einen dritten Dreiklang mit Hilfe der Fourier-Transformation. Führe den folgenden Code aus, um dir den gesuchten Dreiklang anzuhören und im Zeit- und Frequenzraum anzusehen. Bestimme die Frequenzen des Signals und wähle im Dropdownmenü unter der Grafik den zugehörigen Dreiklang aus.



Hier findest du die Übersicht der Dreiklänge.

Wenn du mit dem Mauszeiger über die Graphen fährst, werden dir die Koordinaten (x-Koordinate, y-Koordinate) der gezeichneten Datenpaare angezeigt.

In [6]: *# Hier nichts ändern, nur ausführen!*
`play_unknown_triad3(); plot_unknown_triad3(); check_unknown_triad3()`

▶ 0:00 / 0:02 ⏪ ⏹ ⏷

Dreiklang: C-Dur

Überprüfen

3) Einen Dreiklang nachbilden

Analog zu Aufgabe 2 sollst du auch hier zunächst einen Dreiklang identifizieren. Anschließend wirst du aber nicht bloß dessen Namen angeben, sondern den erkannten Dreiklang selbst noch einmal reproduzieren. Ein Vergleich mit dem ursprünglichen Dreiklang zeigt dir dann, ob du alles richtig gemacht hast.

a) Gesuchter Dreiklang

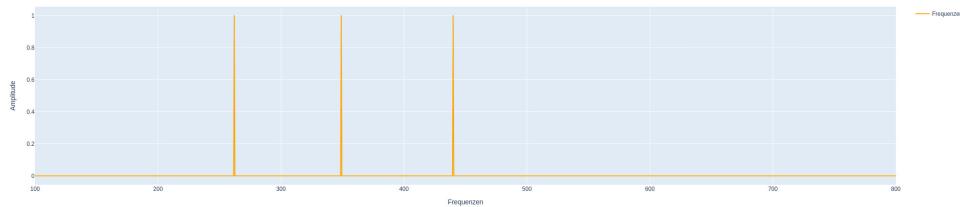
Führe den folgenden Code aus, um dir den gesuchten Dreiklang anzu hören und im Zeit- und Frequenzraum anzusehen.

Wenn du mit dem Mauszeiger über die Graphen fährst, werden dir die Koordinaten (x-Koordinate, y-Koordinate) der gezeichneten Datenpaare angezeigt.

```
In [7]: # Hier nichts ändern, nur ausführen!
play_original_triad()
plot_original_triad()
```

▶ 0:00 / 0:02 ⏸ ⏴ ⏵ ⏴ ⏵

306



b) Dreiklang nachbilden

Nun sollst du den obigen Dreiklang selbst noch einmal erzeugen. Dazu implementierst du zunächst separat die drei einzelnen Sinustöne. Die Reihenfolge spielt dabei keine Rolle. Ersetze im folgenden Code die ersten drei `None` durch die im Dreiklang enthaltenen Sinustöne. Danach bildest du aus diesen Tönen einen Dreiklang. Ersetze dazu das letzte `None` durch den Dreiklang selbst. Klicke anschließend auf "Run". Daraufhin kannst du dir die drei implementierten Töne nacheinander anhören (erst `tone1`, dann `tone2`, zum Schluss `tone3`). Außerdem kannst du dir deren Summe `S3` anhören. Du erhältst 2 Graphen. Der erste zeigt die einzelnen Sinusschwingungen (`tone1` in Blau, `tone2` in Orange, `tone3` in Grün). Der zweite zeigt den Graphen von `S3`. Wenn du alles richtig gemacht hast, entspricht dein Dreiklang dem Original.

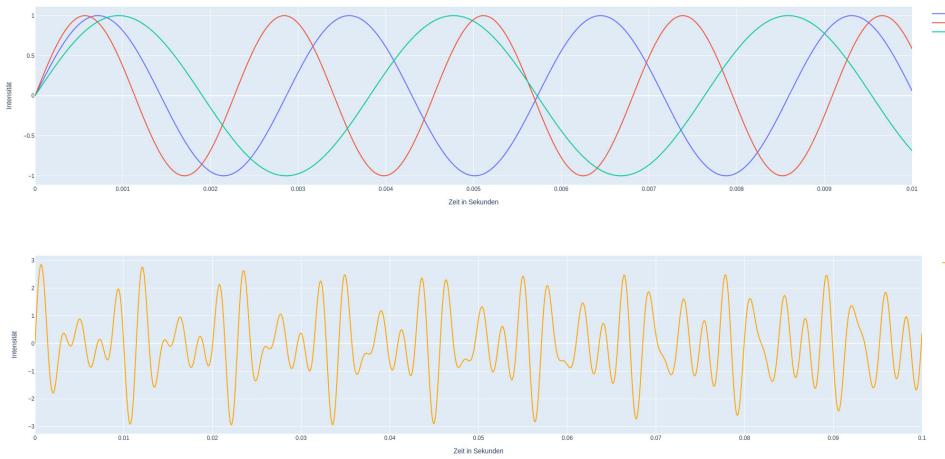
In [8]: *# Setze den obigen originalen Dreiklang aus dessen einzelnen Teiltönen zusammen*

```
def tone1(t):
    # Ersetze 'None' durch den ersten Teilton
    return sin(349 * 2 * pi * t) #= return None=#
def tone2(t):
    # Ersetze 'None' durch den zweiten Teilton
    return sin(440 * 2 * pi * t) #= return None=#
def tone3(t):
    # Ersetze 'None' durch den dritten Teilton
    return sin(262 * 2 * pi * t) #= return None=#
def S3(t):
    # Ersetze 'None' durch den Dreiklang
    return tone1(t) + tone2(t) + tone3(t) #= return None=#
# Ab hier nichts mehr ändern
play_tones(tone1, tone2, tone3, S3)
play_S(S3)
plot_S(tone1, tone2, tone3, S3)
```

✓ Super! Du hast den Dreiklang richtig erkannt und reproduziert.

▶ 0:00 / 0:10 ━━━━ 🔍 ⏮

▶ 0:00 / 0:02 ━━━━ 🔍 ⏮



Fazit

Du hast nun kennengelernt, wie Frequenzen aus Dreiklängen mit Hilfe der Fourier-Transformation ermittelt werden. Auf dem nächsten Arbeitsblatt werden wir erkunden, wie dieses Verfahren vom einfachen Dreiklang auf ein Sprachsignal übertragen werden kann.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 2 gemeinsam im Plenum besprechen.

Öffne nun das nächste [Arbeitsblatt](#).

Schon fertig?

Erfahre auf diesem [Zusatzblatt](#) mehr über die Funktionsweise der Fourier-Transformation und führe die Fourier-Transformation auch einmal eigenhändig durch. Hinweis: Hierzu ist Vorwissen zur Integration notwendig.



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Autorinnen und Autoren: Maike Gerhard und Lars Schmidt

B.3 Arbeitsblatt 3

Arbeitsblatt 3 | Merkmale reduzieren



Auf dem letzten Arbeitsblatt wurde erarbeitet, wie die Frequenzen aus einem Dreiklang ermittelt werden. Wir werden auf diesem Arbeitsblatt kennenlernen, wie dieses Verfahren vom einfachen Dreiklang auf ein Sprachsignal übertragen werden kann. Anschließend erarbeiten wir Methoden, um die ermittelten Frequenzen für die spätere Erkennung des Signals zu reduzieren.

```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern, nur ausführen!
import sys; sys.path.append("../code"); from setup_AB3 import *
```

1) Untersuchung von Tonabfolgen

Hört man sich ein Sprachsignal an, so fällt der Unterschied zum Dreiklang direkt auf. Ein Sprachsignal ist deutlich dynamischer, da die Mischung der Frequenzen nicht wie beim Dreiklang konstant ist, sondern sich mit der Zeit ändert. Mathematisch bedeutet das, dass sich die Frequenzraum-Darstellung eines Sprachsignals (im Gegensatz zu der eines Dreiklangs) im zeitlichen Verlauf ändert. Für die Erkennung eines Sprachsignals müssen wir die Frequenzraum-Darstellung zu jedem Zeitpunkt kennen. Wie man ein komplexes akustisches Signal innerhalb einzelner Zeitblöcke hinsichtlich seiner Frequenzzusammensetzung analysiert, wird in dieser Aufgabe erarbeitet. Als akustisches Signal dient hier zur Vereinfachung die **Abfolge dreier verschiedener Dreiklänge**, an der die Vorgehensweise gut veranschaulicht werden kann. Die erarbeitete Vorgehensweise kann anschließend auch bei komplexeren akustischen Signalen, wie Sprachsignalen, angewendet werden.

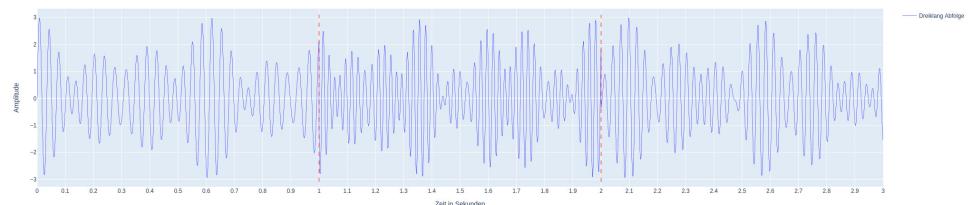


a) Tonabfolge anhören

Hör dir die zu untersuchende Tonabfolge an.

```
In [2]: # Hier nichts ändern!
triad_sequence, frequencies = play_triad_sequence(); plot_triad_sequence()
```

▶ 0:00 / 0:03 ━ ━ ━ ━ :



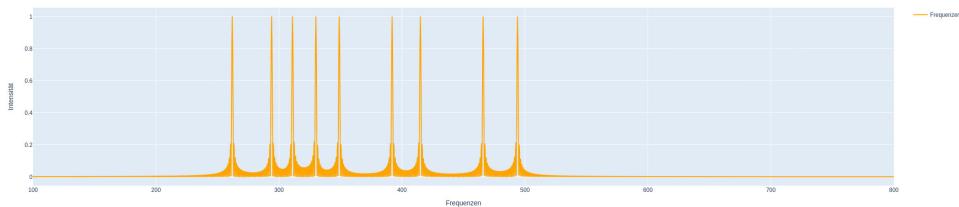
Wir werden nun die neun Frequenzen bestimmen, die in der Dreiklangabfolge enthalten sind. Dazu wenden wir die gleiche Methode, wie zuvor bei der Bestimmung eines unbekannten Dreiklangs, an: die **Fourier-Transformation**.



b) Frequenzen ausgeben

Führe das folgende Codefeld aus, um dir die neun Frequenzen (es handelt sich um drei Dreiklänge) ausgeben zu lassen, die in der Dreiklangabfolge enthalten sind.

```
In [3]: # Hier nichts ändern, nur ausführen!
compute_fourier_analysis(triad_sequence)
```



c) Dreiklang bestimmen

Trage die Frequenzen, die sich aus Aufgabenteil b ergeben, im folgenden Codefeld ein. Gib dafür drei Frequenzen für den ersten Dreiklang, drei für den zweiten und drei für den dritten Dreiklang an.

Höre dir deine erstellte Abfolge an und vergleiche das Audiosignal mit dem Audiosignal aus Aufgabenteil a. Diskutiere mit deinem Sitznachbarn oder deiner Sitznachbarin Gründe für mögliche Unterschiede oder Gemeinsamkeiten der Audiosignale.

```
In [4]: # Trage hier die Frequenzen ein

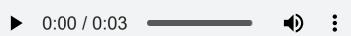
# Erster Dreiklang
frequency1_a = 262 #=None=
frequency1_b = 294 #=None=
frequency1_c = 311 #=None=

# Zweiter Dreiklang
frequency2_a = 330 #=None=
frequency2_b = 349 #=None=
frequency2_c = 392 #=None=

# Dritter Dreiklang
frequency3_a = 415 #=None=
frequency3_b = 466 #=None=
frequency3_c = 494 #=None=

# Hier nichts ändern
play_reconstructed_triad_sequence(frequency1_a, frequency1_b, frequency1_c, frequency2_a, frequency2_b,
```

✓ Du hast die Frequenzen korrekt bestimmt! Hör dir deine Dreiklangabfolge an.



▼ Klicke hier, nachdem du Aufgabenteil c bearbeitet hast.

Es fällt auf, dass das Ergebnis nicht mit der ursprünglichen Dreiklangabfolge übereinstimmt. Das liegt daran, dass am Amplitudenspektrum nicht abzulesen ist, welche drei Frequenzen jeweils zu welchem Zeitpunkt erklingen und damit welchen Dreiklang erzeugen. Um herauszufinden, in welchem Zeitblock welche Frequenzen erklingen, ist es notwendig die Fouriertransformation separat für einzelne Zeitblöcke durchzuführen.

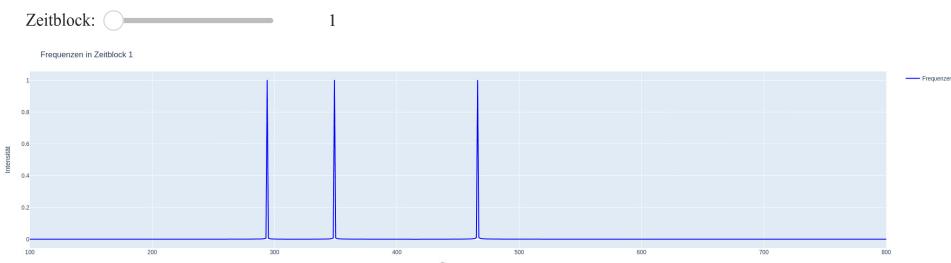


d) Dreiklangabfolge unterteilen

Gib im folgenden Codefeld die Anzahl der Zeitblöcke und die Länge der Zeitblöcke ein, um herauszufinden, ob die fehlende Unterteilung in Blöcke der einzige Grund war, warum die Frequenzbestimmung kein zufriedenstellendes Ergebnis geliefert hat. Führe anschließend das Codefeld aus und bewege den Schieberegler um dir die Frequenzen der verschiedenen Zeitblöcke anzusehen.

```
In [5]: # Aufteilung der Dreiklangabfolge in Blöcke
number_of_blocks = 3 #=None#
# Angabe in Sekunden
length_of_blocks = 1 #=None#
# Hier nichts ändern
block_frequencies = print_frequencies_of_blocks(number_of_blocks, length_of_blocks, triad_sequence)
check_blocks(number_of_blocks, length_of_blocks, triad_sequence)
```

✓ Du hast die Länge der einzelnen Blöcke und die Anzahl der Blöcke korrekt bestimmt.



e) Dreiklänge bestimmen

Im letzten Aufgabenteil wurde die Fourier-Transformation einzeln auf jeden Dreiklang angewendet, sodass die Frequenzen den Amplitudenspektren entnommen werden können. Um welche drei Dreiklänge handelt es sich? Führe das Codefeld aus und wähle den Dreiklang aus.



Hier findest du die Übersicht der Dreiklänge.

```
In [6]: # Hier nichts ändern, nur ausführen!
check_unknown_triad_sequence(block_frequencies)
```

Erster Dreiklang: C-Dur

Zweiter Dreiklang: C-Dur

Dritter Dreiklang: C-Dur

Überprüfen

2) Frequenzen in einem Sprachsignal

Du hast kennengelernt, wie man ein komplexes akustisches Signal innerhalb einzelner Zeitblöcke hinsichtlich seiner Frequenzzusammensetzung analysiert. Diese Vorgehensweise wird nun bei einem realen Sprachsignal angewendet. Hierbei wird das Sprachsignal allerdings nicht nur in 3 Zeitblöcke sondern in 44 Zeitblöcke unterteilt, sodass möglichst viele zeitliche Änderungen in der Frequenzzusammensetzung des Sprachsignals erfasst werden können.



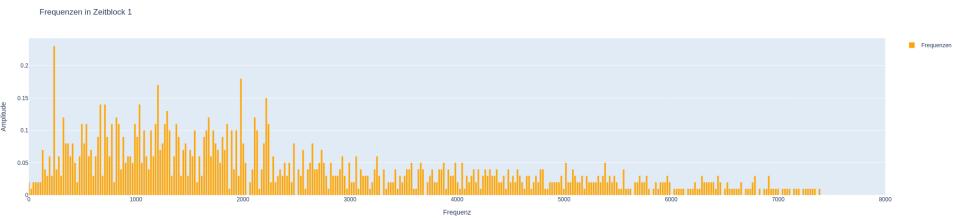
a) Frequenzen bestimmen

Führe das folgende Codefeld aus, um beispielhaft für das gesprochene Wort "two" die Frequenzzusammensetzung ausgeben zu lassen. Bewege den Schieberegler um dir die Frequenzen der verschiedenen Zeitblöcke anzusehen. Notiere im Freitextfeld die Unterschiede, die dir zwischen dem Amplitudenspektrums des Sprachsignals und dem Amplitudenspektrum der Dreiklangabfolge auffallen.

```
In [7]: # Hier nichts ändern!
check_AB3_2a(); prompt_1()
```

► 0:00 / 0:01 ━━ ♪ ⋮

Zeitblock: 1



Deine Antwort: Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern

▼ Klicke hier, nachdem du Aufgabenteil a bearbeitet hast.

Wie du siehst, sind in einem realen Sprachsignal sehr viel mehr Frequenzen überlagert, als bei einer simplen Dreiklangabfolge. Diese Frequenzen müssen bei der Erkennung des Sprachsignals alle weiter verarbeitet werden. Ein großes Problem dabei ist, dass die charakteristischen Teile des Sprachsignals von vielen anderen Frequenzen "zugedeckt" sind. Dies sind Frequenzen, die zum Beispiel nicht zum Sprachsignal gehören, sondern Hintergrundgeräusche sind. In

manchen Bereichen sind die Frequenzen auch unnötig genau aufgelöst. Dies macht die Analyse und Erkennung des Sprachsignals schwierig. Deshalb versucht man die ermittelten Frequenzen auf die Wichtigsten zu reduzieren. Wir bezeichnen diese ausgewählten Frequenzen im Folgenden als **Merkmale** des Sprachsignals.



b) Frequenzen reduzieren

Sammle hier deine Ideen, wie wir die Frequenzen sinnvoll reduzieren können.

In [8]: `# Hier nichts ändern!`
`prompt._2()`

Deine Ideen: Speichern

▼ Klicke hier, nachdem du Aufgabenteil b bearbeitet hast.

i Reduktion auf die wichtigsten Merkmale durch die Orientierung am menschlichen Hörvorgang

Bei der Ermittlung der Merkmale kann man sich am **menschlichen Hörvorgang** orientieren, weil Sprache ideal auf das angepasst ist, was wir als Menschen hören können. Folgende Maßnahmen werden deshalb getroffen:

- Es erscheint logisch die Frequenzen zu streichen, welche vom menschlichen Gehör nicht wahrgenommen werden können.
- Der Mensch kann Frequenzen, die sehr nahe beieinander liegen nicht unterscheiden. Mehrere nebeneinanderliegende Frequenzen werden deshalb zusammengefasst.
- Hohe Frequenzen werden vom menschlichen Gehör weniger differenziert wahrgenommen als niedrige Frequenzen. Die Frequenzen werden daher unterschiedlich stark berücksichtigt.

3) Besonders hohe und besonders niedrige Frequenzen

In akustischen Experimenten wurde festgestellt, dass Menschen Sprache nur in einem bestimmten Frequenzbereich wahrnehmen können. Dieser liegt zwischen 100Hz und 6800Hz .

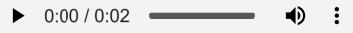


a) Grenzfrequenzen

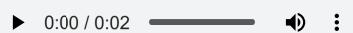
Trage die Grenzfrequenzen im untenstehenden Codefeld ein und führe das Codefeld aus. Du kannst dir anschließend die Grenzfrequenzen der Sprachwahrnehmbarkeit anhören.

```
In [9]: # Ersetze die 'None' durch die untere und obere Grenzfrequenz
frequency_lower = 100 #=None=
frequency_upper = 6800 #=None=

# Ab hier nichts mehr ändern
play_given_tone1(); play_given_tone2();
plot_your_tones(frequency_lower, frequency_upper)
```



untere Grenzfrequenz



obere Grenzfrequenz

✓ Super! Beide Grenzfrequenzen stimmen.



b) Filtern

1.

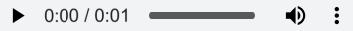
Im Folgenden werden wir Frequenzen unterhalb der unteren Grenzfrequenz und oberhalb der oberen Grenzfrequenz aus einem Sprachsignal entfernen. Trage dazu erneut die Grenzfrequenzen ein und führe den Code aus um dir das Amplitudenspektrum für einen beispielhaften Zeitblock sowie das originale und das gefilterte Signal anzuhören. Vergleiche die beiden Sprachsignale.

2.

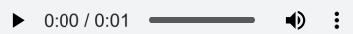
Teste auch andere Werte für die Grenzfrequenz (z.B. mal 1000 Hz als untere Grenzfrequenz und mal 3000Hz als obere Grenzfrequenz). Führe das Codefeld erneut aus und vergleiche die Sprachsignale.

```
In [10]: # Ersetze die 'None' durch die untere und obere Grenzfrequenz
frequency_lower = 100 #=None=
frequency_upper = 6800 #=None=

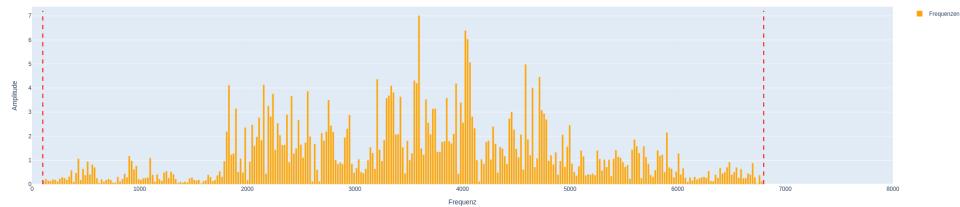
# Ab hier nichts mehr ändern
filter_wav_file("../data/two/00b01445_nohash_0.wav", "tones/filtered_file.wav", frequency_lower, frequency_upper)
```



originales Sprachsignal



gefiltertes Sprachsignal



4) Frequenzen zusammenfassen

Auch nachdem alle besonders hohen und besonders tiefen Frequenzen aus dem Signal entfernt wurden, bleiben noch sehr viele Frequenzwerte übrig. Um diese noch stärker zu reduzieren, orientieren wir uns wieder am menschlichen Gehör:

In Experimenten wurde festgestellt, dass Frequenzen, die sehr nahe beieinander liegen, vom menschlichen Gehör als eine Frequenz aufgefasst werden. Frequenzen die nur einen geringen Frequenzunterschied haben, können von uns Menschen somit gar nicht unterschieden werden.

Im Folgenden werden wir die Frequenzen daher in Frequenzintervallen zusammenfassen.



a) Frequenzintervalle bilden

Diskutiere mit deinem Sitznachbarn oder deiner Sitznachbarin, wie die Amplituden nebeneinanderliegender Frequenzen sinnvoll zu einem einzigen Wert zusammengefasst werden können. Wähle anschließend eine der Antwortmöglichkeiten aus.

```
In [11]: # Hier nichts ändern!
show_correct_text_selection(options1,0)
```

- Den Mittelwert der Amplituden der Frequenzen eines Frequenzintervall bilden.
- Die Summe der Amplituden der Frequenzen eines Frequenzintervall bilden.
- Die größte Amplitude aus dem Frequenzintervall auswählen.
- Die Amplitude der Frequenz, die genau in der Mitte des Frequenzintervall liegt, auswählen.

Überprüfen

Wir werden eine weitere Information zum menschlichen Gehör ausnutzen, um die Reduzierung auf wenige Werte für ein Sprachsignal noch effektiver zu gestalten: Die Frequenzauflösung des menschlichen Gehörs nimmt mit höheren Frequenzen stark ab. Zwei tiefe Frequenzen mit einem kleinen Frequenzabstand können vom Menschen somit besser unterschieden werden als zwei hohe Frequenzen mit dem gleichen Frequenzabstand. Bisher wurden die Frequenzintervalle bei hohen und tiefen Frequenzen gleich breit gewählt. Wie sollten wir diese Wahl anpassen?



b) Breite der Frequenzintervalle anpassen

Entscheide, wie die Breite der Frequenzintervalle gewählt werden soll.

▼ Tipp

Bei einer schlechten Auflösung können viele Frequenzen zusammengefasst werden, da sie ohnehin vom Gehör nicht unterschieden werden können.

In [12]: `# Hier nichts ändern!`
`show_correct_text_selection(options2,1)`

- Mit höheren Frequenzen nimmt die Breite der Frequenzintervalle ab.
- Mit höheren Frequenzen nimmt die Breite der Frequenzintervalle zu.
- Bei niedrigen Frequenzen sind die Frequenzintervalle schmal, werden dann breiter und zu hohen Frequenzen hin wieder schmal.

Überprüfen



Die unter dem Diagramm dargestellte Liste zeigt die 25 Merkmale des ausgewählten Zeitblocks. Auch in den anderen Zeitblöcken werden die Amplituden der Frequenzen mit diesem Vorgehen auf 25 Merkmale reduziert. Die Merkmale aller Zeitblöcke eines Signals können in einer Tabelle, wie unten zu sehen, gespeichert werden. Werden alle charakteristischen Merkmale eines Sprachsignals wie in der Tabelle zusammengekommen, so spricht man von einem Muster.

	Merkmal 1	Merkmal 2	...	Merkmal 25
Merkmale in Zeitblock 1	0.39	0.45	...	1.19
⋮				
Merkmale in Zeitblock 44			...	

Tabelle 1: Muster eines Sprachsignals

5) Rückblick: Merkmalsermittlung

Wir haben verschiedene Schritte kennengelernt um aus einem Sprachsignal Merkmale zu ermitteln und zu reduzieren.



Sortiere die folgenden Schritte für die Ermittlung der Merkmale eines Sprachsignals.

In [13]: `# Hier nichts ändern!`
`show_text_order()`
`#korrekte Reihenfolge: Sprachsignal in Zeitblöcke unterteilen - Amplituden einzelner Frequenzen durch Fourier-Transformation bestimmen - Frequenzen, die nahe beieinanderliegen, zu Frequenzintervallen zusammenfassen - alle ermittelten Merkmale pro Zeitblock als ein Muster zusammenfassen - Sprachsignal in Zeitblöcke unterteilen`

alle ermittelten Merkmale pro Zeitblock als ein Muster zusammenfassen	^	▼
Sprachsignal in Zeitblöcke unterteilen	^	▼
Frequenzen, die nahe beieinanderliegen, zu Frequenzintervallen zusammenfassen	^	▼
Amplituden einzelner Frequenzen durch Fourier-Transformation bestimmen	^	▼
besonders tiefe / hohe Töne entfernen	^	▼

Überprüfen

Fazit

Du hast nun kennengelernt, wie Merkmale aus Sprachsignalen ermittelt werden. Auf dem nächsten Arbeitsblatt werden wir erkunden, wie diese Merkmale genutzt werden können um ein Sprachsignal zu erkennen.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 3 gemeinsam im Plenum besprechen.

Öffne nun das nächste [Arbeitsblatt](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Autorin: Stephanie Hofmann

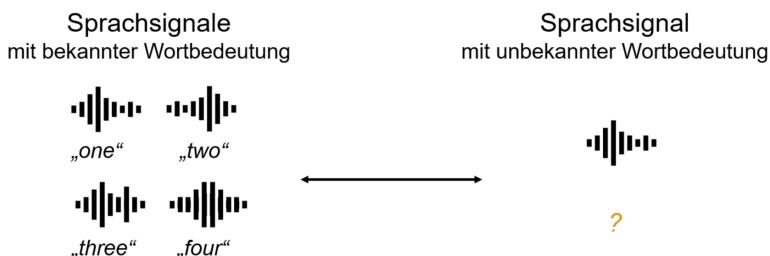
B.4 Arbeitsblatt 4

Arbeitsblatt 4 | Abstände zwischen Merkmalsvektoren



Unser Ziel ist nun, Sprachsignale mit unbekannter Wortbedeutung zu erkennen. Um dies zu vereinfachen, schränken wir die Wortauswahl ein und legen fest, dass es uns genügt, eines der vier Wörter "one", "two", "three", "four" zu erkennen.

Zunächst müssen wir herausfinden welche Merkmale ein Sprachsignal bei jedem dieser vier Wörter typischerweise aufweist. Dazu verwenden wir einen Datensatz, welcher aus Sprachsignalen zu diesen vier Wörtern besteht. Die Wörter wurden von verschiedenen Personen eingesprochen und gemeinsam mit ihrer Wortbedeutung abgespeichert. Aus diesen Sprachsignalen können nun die Merkmale wie auf Arbeitsblatt 1 bis 3 besprochen, ermittelt und als Muster abgespeichert werden. Für jedes Wort werden diese Muster nun als Referenzmuster abgespeichert. Diese Referenzmuster werden nach ihrer Wortbedeutung in die vier Klassen "one", "two", "three", "four" unterteilt.



1) Daten laden

Im folgenden Codefeld wird für unseren Datensatz die Merkmalsermittlung ausgeführt und es werden die verschiedenen Referenzmuster geladen.

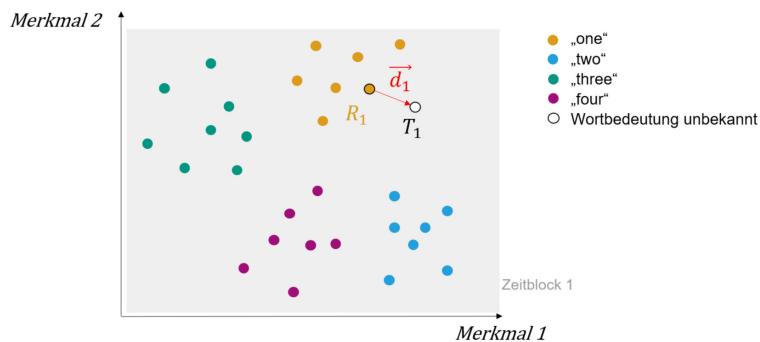
```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern!
import sys; sys.path.append("../code"); from setup_AB4 import *;check_AB4_1();
```

✓ Referenzmuster geladen.

2) Ähnlichkeiten messen

Wollen wir ein neues Sprachsignal erkennen, so können wir dessen Sprachmuster mit den Referenzmustern vergleichen. Wir ordnen das Sprachsignal schließlich der Klasse zu, deren Sprachmuster am ähnlichsten zu unserem neuen Sprachmuster sind.

Wir bezeichnen Sprachmuster als ähnlich, wenn ihr Abstand möglichst gering ist. Dabei vergleichen wir die Sprachmuster Zeitblock für Zeitblock. Pro Zeitblock betrachten wir zunächst nur zwei Merkmale. So können wir diese Merkmale als Punkte in der Ebene auffassen. In der untenstehenden Abbildung wird das Vorgehen für einen Zeitblock schematisch veranschaulicht. Die Punkte sind je nach Klassenzugehörigkeit unterschiedlich eingefärbt. Der Abstand zum weißen umkreisten neuen Punkt, dessen Wortbedeutung noch nicht bekannt ist, soll nun bestimmt werden.



a) Abstand im ersten Zeitblock für 2 Merkmale

Bestimme den Abstand des Punktes von einem Referenzmuster im Zeitblock 1, genannt R_1 und des Punktes des neuen Sprachmusters im Zeitblock 1, genannt T_1 .

☞ **Variablen:** Verwende im Code `R1` für R_1 und `T1` für T_1 . Du musst und sollst hier explizit nicht mit konkreten Werten für R_1 und T_1 arbeiten.

Hinweise:

- Auf die einzelnen Einträge der Vektoren kann mit einer eckigen Klammer und der Nummer des Eintrages zugegriffen werden. Aber Achtung: In Python wird mit dem Zählen bei 0 gestartet. Auf den ersten Eintrag kann also beispielsweise mit `T1[0]` zugegriffen werden.
- Wurzeln \sqrt{x} werden im Code mit `math.sqrt(x)` eingegeben.
- Exponenten wie z.B. x^2 werden im Code mit `x**2` eingegeben.



Tipp

```
In [2]: d_1_a = math.sqrt((R1[0]-T1[0])**2+(R1[1]-T1[1])**2) #=None#
# Ab hier nichts mehr ändern
check_AB4_2a(d_1_a, R1, T1);
```

✓ Deine Formel ist korrekt!



b) Abstand im ersten Zeitblock für 3 Merkmale

Wir werden nun ein weiteres Merkmal hinzunehmen und 3 Merkmale betrachten. Die Merkmale eines Sprachsignals im ersten Zeitblock können wir als Punkte im dreidimensionalen Raum auffassen. Bestimme erneut den Abstand der Punkte R_1 und T_1 . Wie verändert sich die Formel im Vergleich zu Teilaufgabe a)?

☞ **Variablen:** Verwende im Code `R1` für R_1 und `T1` für T_1 .

Hinweise:

- Auf die einzelnen Einträge der Vektoren kann mit einer eckigen Klammer und der Nummer des Eintrages zugegriffen werden. Aber Achtung: In Python wird mit dem Zählen bei 0 gestartet. Auf den ersten Eintrag kann beispielsweise mit `T1[0]` zugegriffen werden.
- Wurzeln \sqrt{x} werden im Code mit `math.sqrt(x)` eingegeben.
- Exponenten wie z.B. x^2 werden im Code mit `x**2` eingegeben.

 Tipp

```
In [3]: d_1_b = math.sqrt((R1[0]-T1[0])**2+(R1[1]-T1[1])**2+(R1[2]-T1[2])**2) #None#
# Ab hier nichts mehr ändern
check_AB4_2b(d_1_b, R1, T1);
```

✓ Deine Formel ist korrekt!



c) Abstand im ersten Zeitblock für 25 Merkmale

Tatsächlich betrachten wir 25 Merkmale pro Zeitblock. Wir können uns die Merkmale nun nicht mehr bildlich als Punkte im Raum vorstellen, aber wir können jetzt auf unsere Vorüberlegungen zurückgreifen und die obige Formel für die Berechnung des Abstandes bei drei Merkmalen erweitern. Notiere deine Formel zunächst auf Papier. Führe anschließend das Codefeld aus und wähle deine entwickelte Formel aus. Entspricht deine Formel keiner der Auswahlmöglichkeiten, so sieh dir den Tipp an.

 Tipp

```
In [4]: # Hier nichts ändern, nur ausführen!
show_correct_text_selection(options_c)
```

$d_1 = \sqrt{(R_1(0) - T_1(0))^2 + (R_1(1) - T_1(1))^2 + \dots + (R_1(24) - T_1(24))^2}$

$d_1 = \sqrt{(R_1(0) - T_1(0))^1 + (R_1(1) - T_1(1))^2 + \dots + (R_1(24) - T_1(24))^2}$

$d_1 = \sqrt{(R_1(0) - T_1(0))^2} + \sqrt{(R_1(1) - T_1(1))^2} + \dots + \sqrt{(R_1(24) - T_1(24))^2}$

Überprüfen

Bisher haben wir lediglich einen einzigen Zeitblock betrachtet und den Abstand zwischen dem neuen Sprachmuster und dem Referenzmuster für diesen Zeitblock berechnet. Tatsächlich wurden die eingesprochenen Signale bei der Fourier-Transformation allerdings in 44 Zeitblöcke unterteilt. Die Abstände d_1, \dots, d_{44} in den weiteren Zeitblöcken können nach dem gleichen Prinzip wie im Zeitblock 1 berechnet werden.

Für den Vergleich der gesamten Sprachmuster ist es notwendig einen Gesamtabstand zu bestimmen.



d) Formel für den Gesamtabstand

Überlegt, welche Formel für den Gesamtabstand d_{ges} am meisten Sinn macht und diskutiert gemeinsam. Wählt anschließend die Formel aus, die für euch am besten passt. Vergleicht die Rückmeldung mit euren Überlegungen.

 **Variablen:** d_1 steht für den Abstand im ersten Zeitblock, d_2 steht für den Abstand im zweiten Zeitblock und so weiter.

In [5]: `# Hier nichts ändern!`
`show_correct_text_selection(options_d)`

- $d_{ges} = d_1 - d_2 - \dots - d_{44}$
- $d_{ges} = d_1 + d_2 + \dots + d_{44}$
- $d_{ges} = d_1 - d_2 + d_3 - \dots - d_{43} + d_{44}$

Überprüfen



e) Gesamtabstand bestimmen

Führe das folgende Codefeld aus. Der Gesamtabstand d_{ges} des Referenzmusters R zum neuen Sprachmuster T wird nach deinen oben angegebenen Formeln berechnet.

In [6]: `# Hier nichts ändern, nur ausführen!`
`check_AB4_2e()`

Der Abstand beträgt 1123 Einheiten.

3) Ein erstes Spracherkennungssystem

Um ein Sprachsignal einer Klasse zuzuordnen wird der Gesamtabstand des zugehörigen Sprachmusters zu jedem Referenzmuster gebildet. Das Sprachsignal wird anschließend der Klasse zugeordnet, welche das Referenzmuster enthält, das den kleinsten Gesamtabstand aufweist.



a) Ein echtes Sprachsignal erkennen

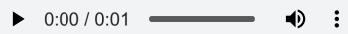
Führe das folgende Codefeld aus um vier beispielhafte Sprachsignale, deren Wortbedeutungen unbekannt sind,

mit unserem Vorgehen zu klassifizieren. Dir wird angezeigt, welcher Klasse das Sprachsignal zugeordnet wird.

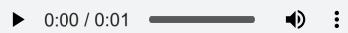
Gib an, wie viele Sprachsignale richtig klassifiziert werden. Benenne mögliche Gründe für eine falsche Klassifikation.

In [7]: `# Hier nichts ändern, nur ausführen!`
`classification_eukl_simple(I_val[3],1)`
`classification_eukl_simple(I_val[0],2)`
`classification_eukl_simple(I_val[17],3)`
`classification_eukl_simple(I_val[24],4)`
`check_AB4_3a()`
`prompt_1()`

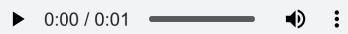
Signal 1 :
Der kleinste Abstand mit 324 Einheiten wurde zu einem Referenzmuster aus Klasse **one** ermittelt.
Das Sprachsignal wird somit als **one** klassifiziert.



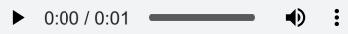
Signal 2 :
Der kleinste Abstand mit 317 Einheiten wurde zu einem Referenzmuster aus Klasse **four** ermittelt.
Das Sprachsignal wird somit als **four** klassifiziert.



Signal 3 :
Der kleinste Abstand mit 476 Einheiten wurde zu einem Referenzmuster aus Klasse **two** ermittelt.
Das Sprachsignal wird somit als **two** klassifiziert.



Signal 4 :
Der kleinste Abstand mit 412 Einheiten wurde zu einem Referenzmuster aus Klasse **two** ermittelt.
Das Sprachsignal wird somit als **two** klassifiziert.



Anzahl der korrekt klassifizierten Sprachsignale:

0

Überprüfen

Mögliche Gründe für eine falsche Klassifikation:
Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern



b) Rückblick: Entwicklung eines Spracherkenners

Sortiere die folgenden Schritte für die Entwicklung eines Spracherkenners in der richtigen Reihenfolge.

In [8]: `# Hier nichts ändern!`
`show_text_order()`

Sprachsignal einlesen	^	▼
Sprachsignal mit der Fourier-Transformation analysieren	^	▼
Klassenzugehörigkeit des Referenzmusters mit kleinstem Abstand auswählen	^	▼
Aus dem Amplitudenspektrum Merkmale ermitteln und Referenzmuster bilden	^	▼
Vergleich des Musters des neuen Sprachsignals mit den Referenzmustern	^	▼

Überprüfen

Fazit

Du hast auf diesem Arbeitsblatt die Klassifikation mit einfacherem Mustervergleich kennengelernt und damit alle Schritte für die Entwicklung eines Spracherkenners durchlaufen. Auf dem nächsten Arbeitsblatt werden wir bewerten, wie gut der entwickelte Spracherkennung funktioniert.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 4 gemeinsam im Plenum besprechen.

Öffne nun das nächste [Arbeitsblatt](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

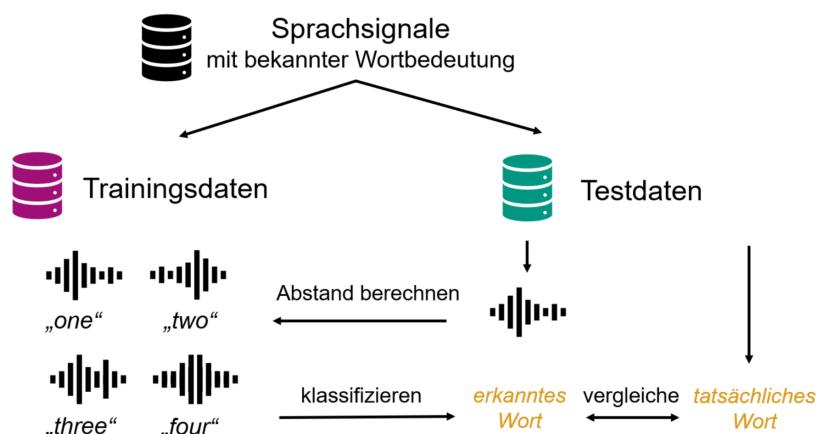
Autorin: Stephanie Hofmann

B.5 Arbeitsblatt 5

Arbeitsblatt 5 | Bewertung des Spracherkenners



Auf Arbeitsblatt 4 haben wir ein erstes funktionierendes Modell für einen Spracherkennner erstellt, indem das Muster eines neuen Sprachsignals mit vorhandenen Referenzmustern abgeglichen wurde. Auf diesem Arbeitsblatt wollen wir bewerten, wie gut der Spracherkennner ist. Wir probieren den Spracherkennner dafür systematisch aus und lesen Sprachsignale ein, deren Wortbedeutung und somit deren Klassenzugehörigkeit bereits bekannt ist. So können wir überprüfen, wie gut die Wortbedeutung über unser entwickeltes Verfahren erkannt wird. Hierfür nutzen wir eine Strategie aus dem Bereich des überwachten Maschinellen Lernens und teilen in einem ersten Schritt unsere Daten in Trainings- und Testdaten auf. Die Trainingsdaten nutzen wir um Referenzmuster zu bilden. Die Testdaten werden zur Bewertung des Spracherkenners genutzt.



1) Die Trainings- und Testdaten

Im Hintergrund haben wir die Daten bereits in Trainings- und Testdaten unterteilt und die Testdaten zurückbehalten. Wir lesen nun sowohl Trainings- als auch Testdaten ein.

Im folgenden Codefeld werden die Referenzmuster, die wir bereits auf Arbeitsblatt 3 aus den **Trainingsdaten** ermittelt haben, sowie deren jeweilige Klassenzugehörigkeit geladen. Außerdem werden die **Testdaten** eingelesen und wie in Arbeitsblatt 3 die zu den Signalen zugehörigen Muster gebildet. Die Klassenzugehörigkeit ist zwar auch bei den Testdaten bekannt, für die Klassifizierung wird diese allerdings noch ausgeblendet und erst später als Vergleichswert bei der Bewertung verwendet.

```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern!
import sys; sys.path.append("../code"); from setup_AB5 import *; check_AB5_1();
```

✓ Referenzmuster und Klassenzugehörigkeit der **Trainingsdaten** geladen. Im Trainingsdatensatz befinden sich 192 Sprachsignale.
Muster der **Testdaten** geladen. Im Testdatensatz befinden sich 48 Sprachsignale.

2) Die Wahrheitstabelle

Wir können die Muster der Testdaten nun dem auf Arbeitsblatt 4 entwickelten Klassifikationsalgorithmus übergeben. Der Algorithmus ordnet jedem Sprachsignal des Testdatensatzes mithilfe der Trainingsdaten eine Wortbedeutung (Klassenzugehörigkeit) zu. Anschließend wird überprüft, ob die über den Klassifikationsalgorithmus bestimmten Wortbedeutungen (Klassenzugehörigkeit) mit den tatsächlichen Wortbedeutungen (Klassenzugehörigkeit) übereinstimmen. Die Ergebnisse können wir uns anschließend in Form einer Tabelle ausgeben lassen. Die Tabelle ist wie folgt aufgebaut:

		Wahrheitstabelle			
		one	two	three	four
tatsächliches Wort	one				
	two				
	three				
	four				
		one	two	three	four
		erkanntes Wort			

Diese Tabelle bezeichnet man auch als **Wahrheitstabelle** (auch Konfusionsmatrix, engl. **confusion matrix**). Sie veranschaulicht die Anzahl der Datenpunkte jeder Klasse, die korrekt bzw. inkorrekt klassifiziert wurden. Falls Datenpunkte falsch klassifiziert wurden, zeigt sie außerdem an, welcher anderen Klasse die Datenpunkte fälschlicherweise zugeordnet wurden.



a) Wahrheitstabelle für den Spracherkennner mit einfachem Mustervergleich

Führe den Code aus. Dir wird die Wahrheitstabelle für die Klassifikation mit dem einfachen Mustervergleich ausgegeben.

```
In [2]: # Hier nichts ändern, nur ausführen!
plot_confusion_matrix(confusion_matrix(classifier_eukl(I_train, I_val)[2], classifier_eukl(I_trai
```

		Wahrheitstabelle			
		one	two	three	four
tatsächliches Wort	one	6	2	0	3
	two	2	6	3	2
	three	0	2	5	1
	four	1	1	1	13
		one	two	three	four
		erkanntes Wort			



b) Interpretation der Wahrheitstabelle

Beantworte die folgenden Fragen und trage deine Antworten in die untenstehenden Felder ein. Sobald du das Häckchen bei "Überprüfen" setzt, werden deine Antworten überprüft.

1. Wie viele Signale wurden insgesamt richtig klassifiziert?
2. Wie viele Signale wurden insgesamt falsch klassifiziert?
3. Welches Wort wurde am häufigsten richtig erkannt?
4. Welches Wort wurde am seltensten richtig erkannt?
5. Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

▼ ⓘ Tipp

zu 1. und 2.: Die jeweilige Anzahl der richtig klassifizierten Signale eines Wortes ist der Diagonaleintrag der Wahrheitstabelle. Alle Werte die nicht auf der Diagonalen eingetragen sind, stehen somit für falsch klassifizierte Datenpunkte.

zu 5.: Unter Überschneidungen verstehen wir Fälle, in denen zwei Wörter verwechselt wurden. Wir suchen die beiden Wörter, die am häufigsten fälschlicherweise für das jeweils andere Wort erkannt wurden.

In [3]: # Hier nichts ändern!
check_AB5_2b()

Frage 1:	0
Frage 2:	0
Frage 3:	"Two" ▾
Frage 4:	"Two" ▾
Frage 5:	"One" und "Three" ▾

Überprüfen

3) Verschiedene Qualitätsmaße

Wir wollen die Ergebnisse mit Hilfe verschiedener **Qualitätsmaße** bewerten.

Zuerst schauen wir uns die **Genauigkeit** (engl. **accuracy**) an. Sie gibt das Verhältnis der richtig klassifizierten Datenpunkte zu allen Datenpunkten an:

$$\text{accuracy} = \frac{\text{Anzahl der richtig klassifizierten Datenpunkte}}{\text{Anzahl aller Datenpunkte}}$$



a) Genauigkeit

Ersetze das `None` durch eine Rechnung, mit der du die Genauigkeit der Klassifikation mit Hilfe der oben ausgegebenen Wahrheitstabelle berechnen kannst. Führe anschließend den Code aus.

☞ Die Eingabe des Geteiltzeichens erfolgt über einen Schrägstrich `/`.

▼ ☰ Tipp 1

Für die Genauigkeit musst du die Anzahl der richtig klassifizierten Datenpunkte bestimmen. Überlege dir dazu, an welcher Stelle in der Wahrheitstabelle die Anzahl der richtig klassifizierten Signale eines Wortes zu finden ist.

▼ ☰ Tipp 2

Die jeweilige Anzahl der richtig klassifizierten Signale eines Wortes ist der Diagonaleintrag der Wahrheitstabelle. Mit Hilfe aller Diagonaleinträge kannst du die Anzahl aller richtig klassifizierten Datenpunkte bestimmen.

```
In [4]: accuracy = (6+6+5+13)/48 #=None=#
# Ab hier nichts mehr ändern
check_AB5_3a(accuracy);
```

✓ Richtig! Die Genauigkeit ist 0.62.

Als weiteres Qualitätsmaß werden wir uns die **Präzision** (engl. **precision**) anschauen. Sie gibt das Verhältnis der richtig als Wort z klassifizierten Datenpunkte zu allen als Wort z klassifizierten Datenpunkten an:

$$\text{precision}_z = \frac{\text{Anzahl der richtig als Wort } z \text{ klassifizierten Datenpunkte}}{\text{Anzahl aller als Wort } z \text{ klassifizierten Datenpunkte}}$$

Falls die Anzahl aller als Wort z klassifizierten Datenpunkte Null ist, gilt $\text{precision}_z = 0$.



b) Präzision

Ersetze die `None` durch Rechnungen, mit denen du die Präzision der einzelnen Wörter mit Hilfe der oben ausgegebenen Wahrheitstabelle berechnen kannst. Führe anschließend den Code aus.

☞ Die Eingabe des Geteiltzeichens erfolgt über einen Schrägstrich `/`.

▼ ☰ Tipp 1

Überlege dir, an welchen Stellen in der Wahrheitstabelle die Anzahl der richtig als Wort z klassifizierten Datenpunkte sowie die Anzahlen aller als Wort z klassifizierten Datenpunkte zu finden sind.

▼  Tipp 2

Die Anzahl der richtig als Wort z klassifizierten Signale ist der Diagonaleintrag des Wortes z . Die Anzahlen aller als Wort z klassifizierten Signale sind die Einträge der Spalte des Wortes z .

```
In [5]: precision_one = 6/(6+2+0+1) #=None=#
precision_two = 6/(2+6+2+1) #=None=#
precision_three = 5/(0+3+5+1) #=None=#
precision_four = 13/(3+2+1+13) #=None=#
# Ab hier nichts mehr ändern
check_AB5_3b(precision_one, precision_two, precision_three, precision_four);
```

✓ Richtig! Die Präzision für Klasse 'one' ist 0.67. Die Präzision für Klasse 'two' ist 0.55. Die Präzision für Klasse 'three' ist 0.56. Die Präzision für Klasse 'four' ist 0.68.



c) Interpretation der Qualitätsmaße

Beantworte die folgenden Fragen und trage deine Antworten in das untere Freitextfeld ein:

1. Was bedeutet es, wenn der Wert der Präzision nahe 0 bzw. nahe 1 liegt? Bei welchen Wörtern ist dies der Fall?
2. Sind die Ergebnisse der Klassifikation zufriedenstellend? Falls nein, wo müssten die Ergebnisse noch verbessert werden?

```
In [6]: # Hier nichts ändern!
prompt_1()
```

Deine Antworten:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

 Speichern

4) Verschiedene Spracherkennner und deren Qualitätsmaße im Vergleich

Uns liegen Testergebnisse eines anderen Spracherkenners vor. Dieser wurde auf einem neuen Testdatensatz getestet. 34 Mal wurde hierbei das Wort "one" richtig erkannt. Das Wort "two" kam 8 Mal, das Wort "three" 3 Mal und das Wort "four" 3 Mal im Testdatensatz vor. Sie wurden jedes Mal fälschlicherweise dem Wort "one" zugeordnet.



a) Wahrheitstabelle erstellen

Trage die Testergebnisse an den richtigen Stellen in der Wahrheitstabelle ein.

▼  Tipp

Über das tatsächliche Wort kann die Zeile und über das erkannte Wort die Spalte des Eintrages in der Wahrheitstabelle ermittelt werden. Die jeweilige Anzahl der richtig klassifizierten Signale eines Wortes wird in der Diagonalen eingetragen.

In [7]: `# Hier nichts ändern!`
`check_AB5_a()`

		one	two	three	four
tatsächliches Wort	one				
	two				
	three				
	four				
		one	two	three	four
		erkanntes	Wort		

 Speichern

Überprüfen



b) Rückschlüsse auf den Spracherkennner

Kannst du aufgrund der Wahrheitstabelle Rückschlüsse auf die Funktionsweise des Spracherkenners ziehen?
Notiere deine Antwort im unteren Freitextfeld.

In [8]: `# Hier nichts ändern!`
`prompt_2()`

Deine Antworten: `Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.`

 Speichern



c) Bestimmung der Qualitätsmaße für diesen Spracherkennner

Berechne die oben eingeführten Qualitätsmaße Genauigkeit und Präzision. Ersetze hierfür die `None` im untenstehenden Code und führe den Code aus.



Die Eingabe des Geteiltzeichens erfolgt über einen Schrägstrich `/`.

In [9]: `accuracy2 = 34/(34+8+3+3) #=None=#
precision_one2 = 34/(34+8+3+3) #=None=#
precision_two2 = 0 #=None=#
precision_three2 = 0 #=None=#
precision_four2 = 0 #=None=#

Ab hier nichts mehr ändern
check_AB5_4c(accuracy2, precision_one2, precision_two2, precision_three2, precision_four2);`

✓ Richtig! Die Genauigkeit ist 0.71 Die Präzision für Klasse 'one' ist 0.71. Die Präzision für Klasse 'two' ist 0. Die Präzision für Klasse 'three' ist 0. Die Präzision für Klasse 'four' ist 0.



d) Vergleich der Spracherkennner

Begründe, welchen Spracherkennner du auswählen würdest.

In [10]: `# Hier nichts ändern!`
`prompt_3()`

Deine Antworten:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern



e) Aussagekraft der Qualitätsmaße

Erkläre, was deine Wahl in Aufgabenteil d) für die Aussagekraft der Qualitätsmaße bedeutet. Notiere deine Antwort im unteren Freitextfeld.

In [11]: `# Hier nichts ändern!`
`prompt_4()`

Deine Antworten:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern

5) Ideen zur Verbesserung

Die Ergebnisse unserer Klassifikation können noch verbessert werden. Daher wollen wir unser Vorgehen überdenken.



Sammle Ideen, was wir an unserem bisherigen Vorgehen verändern könnten, um bessere Ergebnisse zu erhalten. Notiere deine Überlegungen im unteren Freitextfeld.

In [12]: `# Hier nichts ändern!`
`prompt_5()`

Deine Ideen:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern

Fazit

Du hast auf diesem Arbeitsblatt die Güte der Klassifikation mit einfachem Mustervergleich bewertet. Auf dem nächsten Arbeitsblatt werden wir versuchen die Ergebnisse unseres Spracherkenners durch Veränderungen an unserem bisherigen Vorgehen zu verbessern.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 5 gemeinsam im Plenum besprechen.

Öffne nun das nächste [Arbeitsblatt](#).



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Autorin: Stephanie Hofmann

B.6 Arbeitsblatt 6

Arbeitsblatt 6 - Modellverbesserung: Zeitliche Anpassung der Sprachmuster



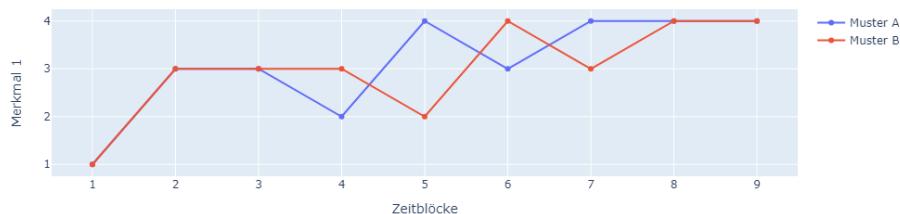
Auf Arbeitsblatt 1-5 haben wir einen Spracherkennner mit einer Klassifizierung über einen einfachem Mustervergleich entwickelt und bewertet. Bei der Klassifizierung hatten wir den Abstand verwendet, um die Ähnlichkeit zweier Sprachmuster zu messen. Auf diesem Arbeitsblatt wollen wir nun das Modell verbessern und die Tatsache, dass verschiedene Laute bei ein und derselben Äußerung unterschiedlich schnell gesprochen werden, berücksichtigen.

```
In [1]: import plotly.io as pio
pio.renderers.default = 'png'
pio.kaleido.scope.default_width = 2048
pio.kaleido.scope.default_height = 512
# Hier nichts ändern!
import sys; sys.path.append("../code"); from setup_AB6 import *
```

Importing the dtw module. When using in academic works please cite:
T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.
J. Stat. Soft., doi:10.18637/jss.v031.i07.

1) Probleme dem bisherigen Klassifizierungsverfahren

Nehmen wir an, es liegen zwei Sprachmuster vor, deren Merkmale für jeden Zeitblock bekannt sind. In der unten abgebildeten Grafik ist für beide Sprachsignale das erste Merkmal jedes Zeitblocks aufgetragen.



Abstand vs. Ähnlichkeit

Berechne den Abstand in den einzelnen Zeitblöcken und addiere diese zu einem Gesamtabstand. Vergleiche den Wert des Gesamtabstandes mit der optischen Ähnlichkeit der beiden Sprachmuster. Notiere deine Erkenntnisse.

```
In [2]: # Hier nichts ändern!
prompt_1()
```

Deine Antworten:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern

▼ Klicke hier, sobald du deine Antwort notiert hast.

Die beiden Sprachsignale weisen optisch sehr ähnliche Muster auf. Muster B scheint ab Zeitblock 4 lediglich um einen Zeitblock gegen Muster A verschoben zu sein. Berechnen wir allerdings den Gesamtabstand zwischen den

Mustern wie auf Arbeitsblatt 4 und gehen davon aus, dass uns nur das erste Merkmal vorliegt, so ergibt sich aus der Summe der Abstände in den einzelnen Zeitblöcken der Gesamtabstand zu $0+0+0+1+2+1+1+0+0 = 5$. Dies erscheint recht viel, wenn man bedenkt, wie ähnlich sich die Sprachmuster sind.

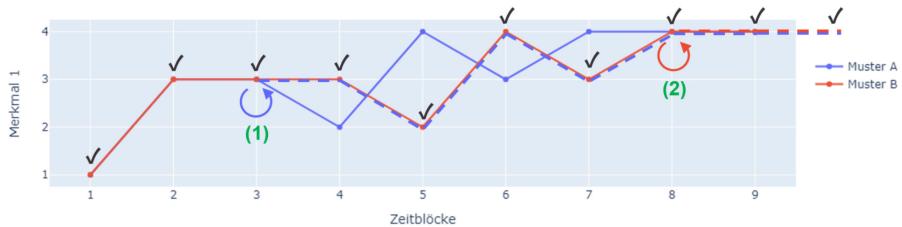
Wir wollen daher die zeitlichen Verschiebungen der Sprachsignale zueinander berücksichtigen. Dazu betrachten wir die Sprachmuster nicht mehr nur Zeitblock für Zeitblock einzeln, sondern nehmen den zeitlichen Verlauf in den Blick.

2) Dynamische Zeitanpassung

Um die Ähnlichkeit zweier Sprachmustern sinnvoll zu erfassen, wollen wir sie nicht einfach übereinander legen und den Gesamtabstand bestimmen, wie wir es bisher getan haben. Stattdessen sollen zeitlich zueinander passende Bereiche in den Mustern übereinander gelegt und anschließend der Gesamtabstand bestimmt werden. Da die zeitlichen Verschiebungen und Verzerrungen nicht nur die gesamte Äußerung, sondern, wie oben im Beispiel, auch nur Teile davon betreffen können, lässt sich dies nicht einfach durch eine globale Stauchung oder Streckung des Sprachmusters erreichen. Vielmehr müssen die einzelnen Teile des Sprachmusters individuell aufeinander angepasst werden. Erforderlich sind daher mehrere lokale Zeitverzerrungen.

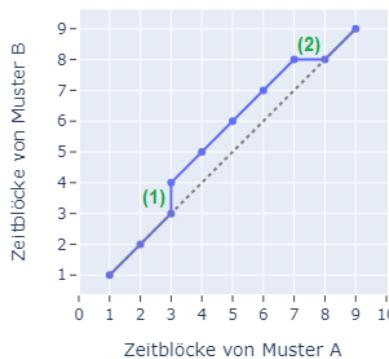
In unserem Beispiel würden folgenden Zeitverzerrungen die Muster aufeinander anpassen:

- Eine zeitliche Streckung des Musters A im Zeitblock 3 - also ein Verweilen im Zustand des Zeitblocks 3 in Muster A für einen weiteren Zeitblock bei einem gleichzeitigen Voranschreiten in Muster B - sorgt dafür, dass die Muster bis Zeitblock 8 perfekt übereinander liegen. (1)
- Eine zeitliche Streckung des Musters B im Zeitblock 8 - also ein Verweilen in Muster B bei einem gleichzeitigen Voranschreiten in Muster A - stellt sicher, dass die Muster gleich lang bleiben. (2)



Diese zeitliche Anpassung lässt sich durch eine Zuordnung der Zustände von Muster A zu den Zuständen von Muster B beschreiben. (1,1) bedeutet Zustand 1 von Muster A wird Zustand 1 von Muster B zugeordnet. Die Zuordnung würde somit wie folgt aussehen:

(1,1), (2,2), (3,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,8), (9,9)
Diese Zuordnung kann auch grafisch als Pfad dargestellt werden:



Mit der gestrichelten Linie ist die Zuordnung dargestellt, bei welcher die Muster ohne zeitliche Verschiebung aufeinander gelegt werden.

Wir betrachten zwei weitere Muster: Muster C und Muster D.

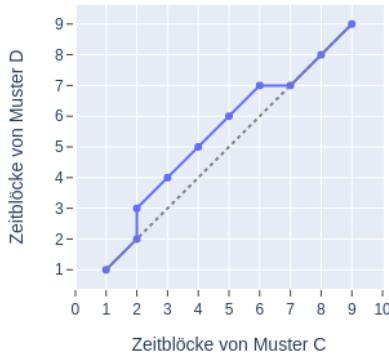


a) Zuordnung festlegen

Bestimme eine Zuordnung für die zeitliche Anpassung der Muster C und D, sodass die einzelnen Teile der Muster möglichst gut aufeinander passen. Trage die Zuordnung im unteren Codefeld ein. Überlege anschließend, ob es noch weitere sinnvolle Zuordnungen gibt. Ersetze die ursprüngliche Zuordnung durch die neue Idee und prüfe deine Eingabe.

```
In [3]: path = [(1,1),(2,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,7),(8,8),(9,9)] #[(1,1),(2,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,7),(8,8),(9,9)]
# Ab hier nichts mehr ändern
check_AB6_2a(path)
```

✓ Richtig! Die Zuordnung passt die Muster zeitlich perfekt aneinander an.



Hier ist eine sinnvolle Zuordnung noch leicht ersichtlich. Trotzdem ergeben sich bereits mehrere Möglichkeiten für eine sinnvolle zeitliche Anpassung.

Betrachten wir die Muster unserer realen Sprachsignale mit 44 Zeitblöcken, so ist es hier wesentlich schwieriger eine Zuordnung zu finden, die die Muster zeitlich gut aufeinander anpasst. Es gibt viele verschiedene Möglichkeiten und es ist nicht klar, welche diejenige ist, die die Muster am besten aufeinander anpasst. Ob die Muster gut aufeinander

passen, können wir am einfachsten über den Abstand der angepassten Muster feststellen. Wir suchen somit diejenige Zuordnung, bei welcher der Abstand der angepassten Muster minimal wird. Um entscheiden zu können, welche Zuordnung den minimalen Abstand liefert, sollten wir zunächst festlegen, welchen Randbedingungen die Zuordnung genügen sollte.

Randbedingungen

Die Randbedingungen der Zuordnung sollen die **Eigenschaften von gesprochener Sprache** wiederspiegeln. Dazu zählen

1. Die zeitliche Reihenfolge der Laute einer Äußerung soll erhalten bleiben, sodass nicht plötzlich eine vom ursprünglichen Signal verschiedene Aneinanderreihung der Laute stattfinden kann und ein komplett neues Wort entsteht. Eine Änderung der zeitlichen Reihenfolge darf somit nicht möglich sein.
2. Keine größeren Teile des Musters und damit z.B. ein ganzer Laut dürfen übersprungen werden.
3. Die beiden Sprachsignale starten und enden zum gleichen Zeitpunkt.



b) Mathematisierung der Randbedingungen

Die Randbedingungen der Zuordnung sollen die Eigenschaften von gesprochener Sprache wiederspiegeln. Wähle aus, welche Randbedingung zu welcher Eigenschaft von gesprochener Sprache passt und trage die zugehörige Zahl ein. Ordne außerdem die Gegenbeispiele zu.

▼ Zusatzaufgabe

Wie viele verschiedene Zuordnungen gibt es bereits bei einer Musterlänge von nur 3 Zeitblöcken und den gewählten Randbedingungen? Wie viele verschiedene Zuordnungen gibt es bei 9 Zeitblöcken?

▼ Tipp

Starte zunächst mit der Beantwortung der ersten Frage. Überlege dir eine Strategie, wie du die unterschiedlichen Zuordnungspfade zu Gruppen mit ähnlichen Eigenschaften zusammenfassen kannst, damit du keinen Zuordnungspfad vergisst.

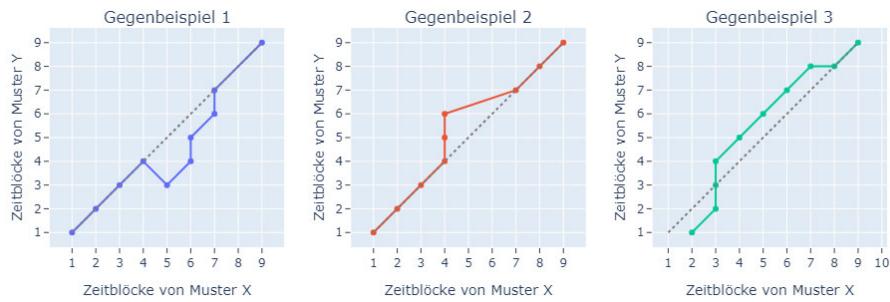
▼ Tipp

Beispielsweise gibt es einen Zuordnungspfad mit zwei Diagonalen. Wie viele Zuordnungspfade gibt es mit einer oder keiner Diagonalen?

Randbedingungen:

- A.** Die Zuordnung startet mit (1,1) und endet mit (9,9).
- B.** Der Zuordnungspfad ist monoton wachsend. Die Merkmale eines Musters müssen somit nacheinander verwendet werden. Sobald ein Merkmal eines Musters eingesetzt wurde, sind nur noch das gleiche oder spätere Merkmale zu verwenden.
- C.** Kein Merkmal darf ausgelassen werden.

Gegenbeispiele:



```
In [4]: # Hier nichts ändern!
check_conditions(); check_examples();
```

Eigenschaft 1:	Randbedingung A	▼
Eigenschaft 2:	Randbedingung A	▼
Eigenschaft 3:	Randbedingung A	▼

✓ Überprüfen

Eigenschaft 1:	Gegenbeispiel 1	▼
Eigenschaft 2:	Gegenbeispiel 1	▼
Eigenschaft 3:	Gegenbeispiel 1	▼

✓ Überprüfen

Demnach sind in jedem Zeitblock drei Richtungen möglich, um den Zuordnungspfad zu bilden: ein Schritt nach oben ↑, nach rechts → oder diagonal ↗.

Eine Möglichkeit die Zuordnung mit dem kleinsten Gesamtabstand zu finden, ist alle möglichen Zuordnungen auszuprobieren. Jedes Mal müssen dabei die Sprachmuster gemäß der Zuordnung aufeinander angepasst und anschließend der Gesamtabstand zwischen den angepassten Sprachmustern ermittelt werden. Trotz der Einschränkungen durch die Randbedingungen wären das allerdings sehr viele Möglichkeiten für die Zuordnung, die ausprobiert werden müssten. Deshalb werden wir jetzt ein effizienteres Verfahren kennenlernen, mit dem wir die Zuordnung finden können, die den geringsten Gesamtabstand liefert.

Verfahren zum Finden der Zuordnung mit minimalem Gesamtabstand

Wir entwickeln das Verfahren am Beispiel der unten abgebildeten Muster E und F.



Hier ist es nur noch schwer möglich, die Zuordnung mit dem minimalen Abstand durch bloßes Hinsehen zu finden. Wir arbeiten uns deshalb Zeitblock für Zeitblock vor, indem zuerst mögliche zeitliche Anpassungen der Sprachmuster in diesem Zeitblock in den Blick genommen werden und anschließend für jede Anpassung der Abstand der angepassten Sprachmuster bestimmt wird:

Schritt 1:

Wir wissen, dass jeder Zuordnungspfad bei (1,1) startet. Im ersten Zeitblock beträgt der Abstand zwischen den Sprachmustern E und F somit

$$|E(1)-F(1)| = |3-4| = 1.$$

Schritt 2:

Im zweiten Zeitblock ergeben sich bereits drei mögliche Zuordnungen: nach oben zu (1, 2) ↑, nach rechts zu (2, 1) → oder diagonal zu (2, 2) ↗ gehen.

Im ersten Fall, wenn wir den Weg nach oben nehmen, ergibt sich die Gesamtabstand bis Zeitblock 2 aus der Summe aus dem Abstand im Zeitblock 1 und dem Abstand in Zeitblock 2 zu

$$|E(1)-F(1)| + |E(1)-F(2)| = 1 + |4-1| = 1 + 3 = 4.$$

Den Abstandswert, der sich beim Weg nach rechts ergibt, sollst du selbst bestimmen.



c) Abstandswerte in Zeitblock 2

Berechne den Gesamtabstand bis Zeitblock 2, wenn wir den Weg nach rechts nehmen.

 **Hinweis:** Du kannst eine Rechnung eintragen (Der Betrag eines Wertes wird im Code mit `abs(Wert)` eingegeben) oder direkt den berechneten Wert eintragen.

```
In [5]: distance = 1 + abs(3-1)
# Ab hier nichts mehr ändern
check_AB6_2c(distance)
```

✓ Richtig! Der Abstand ist 3.

Da mit jedem weiteren Zeitblock mehr Zuordnungen möglich sind und wir damit immer mehr Abstandswerte erhalten, speichern wir die Abstandswerte übersichtlich in einer Tabelle (siehe rechts). Am unteren Rand der Tabelle sind die Merkmale des Musters E und am linken Rand die Merkmale des Musters F eingetragen. Für jeden Punkt des möglichen Zuordnungspfades ist ein Feld für den bis dahin aufsummierten Abstand vorgesehen. In der nebenstehenden Tabelle sind bereits die eben bestimmten Abstandswerte eingetragen.

Der Abstandswert bei (2,2) ist noch nicht eingetragen. Der Zustand (2,2) lässt sich nämlich nicht nur

- diagonal von (1,1) aus (grün)
- sondern auch über die Folge (1, 1), (1, 2), (2, 2) (blau)
- und über die Folge (1, 1), (2, 1), (2, 2) (orange)

erreichen (siehe Tabelle rechts).

Da wir nur an derjenigen Zuordnung interessiert sind, welche den minimalen Gesamtabstand hat, suchen wir den Weg aus, über den bisher der geringste Gesamtabstand aufgesammelt wurde. Hier ist die Zuordnung (1,1), (2,2), also der diagonale Weg, diejenige, welche den geringsten Gesamtabstand aufweist. Der Abstandswert in Zustand (2,2) berechnet sich somit zu

$$|E(1)-F(1)| + |E(2)-F(2)| = 1 + |1-1| = 1 + 0 = 1.$$

3				
3				
1	4			
3	1	3	3	
F/E	4	1	1	3

Schritt 3:

Anhand dieses Schemas werden nun alle weiteren Abstandswerte bestimmt, indem wir zurückblicken: Von welcher Richtung aus sollte ich gekommen sein, sodass der Gesamtabstand bis zu diesem Zeitblock minimal ist? Dabei habe ich die Möglichkeit von unten 1, von links → oder von diagonal unten ↗ gekommen zu sein. Nachdem die Richtung ausgewählt worden ist, kann der neue Abstandswert aus der Summe des Abstandes im aktuellen Zustand und dem vorherigen Abstandswert bestimmt werden.

3				
3				
1	4	↗		
3	1	1	3	
F/E	4	1	1	3

So ergibt sich der Zuordnungspfad und zugleich wird direkt der Gesamtabstand zwischen den zeitlich aufeinander angepassten Sprachmuster bestimmt. Der Wert ganz rechts oben bei (9,9) gibt den Gesamtabstand der beiden Sprachmuster aus. Da uns letztlich nur der Abstand interessiert, werden wir im Folgenden den Zuordnungspfad nicht mehr mitzeichnen, sondern nur noch die Abstandswerte eintragen.



d) die beste Zuordnung finden

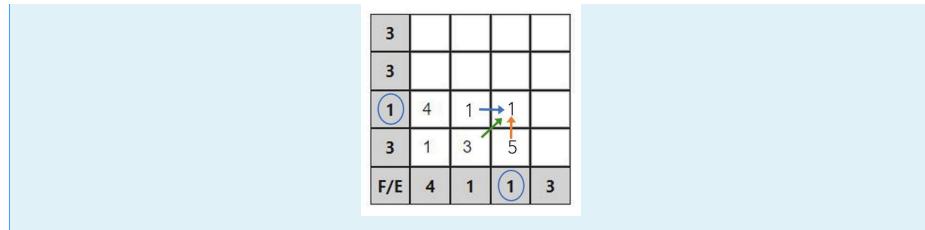
Vervollständige die Tabelle nach dem eben kennengelernten Prinzip und trage die Abstandswerte ein. Klicke auf "Speichern" und anschließend auf "Prüfen" um deine Ergebnisse zu überprüfen.

▼ Tipp

Um den Abstandswert bei (3,1) zu berechnen betrachten wir die möglichen Zuordnungen, die zu diesem Zustand führen. Da von unten und von diagonal unten nicht möglich ist, können wir nur von links gekommen sein. Der Abstandswert ergibt sich somit aus dem Abstandswert links von (3,1) plus dem Abstand in Zustand (3,1). Der Abstand in Zustand (3,1) kann direkt mit Hilfe der Randwerte bestimmt werden: $|3-1|=2$. Somit ergibt sich der Abstandswert zu $3 + 2 = 5$.

3				
3				
1	4	1		
3	1	3	5	
F/E	4	1	1	3

Um den Abstandswert bei (3,2) zu bestimmen, müssen der Zuordnungspfad von links kommend, von diagonal kommend und von unten kommend betrachtet werden. Den kleinsten Abstandswert weist die Zuordnung von links kommend auf. Wir entscheiden uns daher für diesen Zuordnungspfad und addieren den Abstandswert von Zustand (3,2). Der Abstandswert ergibt sich zu $1 + |1-1| = 1$.



In [6]: `# Hier nichts ändern!`
`check_AB6_2d()`

3						Speichern
3						Überprüfen
1	4					
3	1	3				
F/E	4	1	1	3		

❶ Hintergrundinformation zum Verfahren

Dieses kennengelerntes Verfahren wird auch als **Dynamic Time Warping** (DTW) bezeichnet. Die ersten beiden Wörter beziehen sich hierbei auf die dynamische zeitliche Anpassung der Muster. Das Wort "Warping" weist auf die Art und Weise hin, wie sich der Zuordnungspfad über das Koordinatensystem "windet". Das Dynamic Time Warping ist eine Spezialisierung des **Dynamic Programming**. Dieses Verfahren wurde im Jahr 1940 vom Mathematiker Richard Bellman entwickelt, um Minima/ Maxima komplexer Probleme zu finden. Sein Ansatz besteht darin, das Problem in kleinere Teilprobleme aufzuteilen und systematisch Zwischenergebnisse zu speichern.

3) Spracherkennung mit Dynamic Time Warping

Die Klassifizierung der Sprachsignale kann nun genauso, wie auf Arbeitsblatt 4 erarbeitet, stattfinden. Um ein neues Sprachsignal zu klassifizieren wird mit dem DTW Verfahren das zugehörige Sprachmuster an jedes Referenzmuster angepasst und der jeweilige Abstand bestimmt. So kann das Referenzmuster ermittelt werden, welches nach der zeitlichen Anpassung den kleinsten Abstand hat. Die Klasse dieses Referenzmusters wird ausgewählt.



a) Klassifizierungsergebnisse

Führe das folgende Codefeld aus um die Test- und Trainingsdaten aus Arbeitsblatt 5 zu laden und die Klassifizierung der Sprachsignale mit dem DTW Verfahren durchzuführen.

☒ Hier musst du dich etwas gedulden, da die Berechnung der Gesamtabstände aller Sprachmuster im Datensatz etwas länger dauert.

In [7]: `# Hier nichts ändern, nur ausführen!`
`plot_confusion_matrix(confusion_matrix(classifier_dtw(I_train, I_val)[2], classifier_dtw(I_train,`

		Wahrheitstabelle			
		one	two	three	four
tatsächliches Wort	one	9	2	0	0
	two	0	11	1	1
	three	0	2	5	1
	four	0	0	0	16
		one	two	three	four



b) Qualitätsmaße

Trage die Rechnung, mit der du die Genauigkeit der Klassifizierung und die Präzision der vier Klassen mit Hilfe der oben ausgegebenen Wahrheitstabelle berechnen kannst. Führe anschließend den Code aus.

In [8]: `accuracy = (9+11+5+16)/48 #= None =#`

```
precision_one = 9/(9+0+0+0) #= None =#
precision_two = 11/(2+11+2+0) #= None =#
precision_three = 5/(0+1+5+0) #= None =#
precision_four = 16/(0+1+1+16) #= None =#

# Ab hier nichts mehr ändern
check_AB6_3b(accuracy, precision_one, precision_two, precision_three, precision_four)
```

✓ Richtig! Die Genauigkeit ist 0.85. Die Präzision für Klasse 'one' ist 1.0. Die Präzision für Klasse 'two' ist 0.73. Die Präzision für Klasse 'three' ist 0.83. Die Präzision für Klasse 'four' ist 0.89.



c) Vergleich der Spracherkennner

Vergleiche die Werte der Qualitätsmaße des Spracherkenners mit DTW mit den Werten der Qualitätsmaße des Spracherkenners ohne DTW. Begründe, welchen Spracherkennner du auswählen würdest.

In [9]: `# Hier nichts ändern!`
`prompt_2()`

Deine Begründung:

Gib hier deine Antwort ein und bestätige anschließend deine Eingabe.

Speichern



d) Die Spracherkennung selbst ausprobieren

Probiere die Spracherkennung selbst aus. Sprich dazu eines der vier Zahlenwörter ein. Führe dafür das Codefeld aus und drücke den Button "Aufnahme starten". Anschließend startet ein Countdown. Beginne direkt nach dem Countdown mit dem Sprechen, denn die Aufnahme ist nur eine Sekunde lang. Du kannst dein aufgenommenes Sprachsignal mit dem Button "Aufnahme speichern und anhören" abspeichern und es dir anhören, um zu überprüfen, ob deine Äußerung auf der Aufnahme gut zu hören ist. Drücke danach den Button "Sprachsignal erkennen" damit die Spracherkennung durchgeführt wird. Teste auch, ob es eine Auswirkung hat, wie weit du bei der Aufnahme vom Laptop entfernt bist und/ oder wie viele Hintergrundgeräusche mit aufgenommen werden. Führe das Codefeld erneut aus, um ein weiteres Sprachsignal aufzunehmen.

In [10]: `show_timed_audio_recorder("tones/studentsvoice.wav", duration=1)`

Aufnahme starten

Aufnahme speichern und anhören

Sprachsignal erkennen

Fazit

Auf diesem Arbeitsblatt hast du das Verfahren des Dynamic Time Warping für die zeitliche Anpassung zweier Sprachsignale aneinander erarbeitet und auf den Testdatensatz angewendet. Außerdem hast du die Spracherkennung mit und ohne Dynamic Time Warping verglichen.

Diskussion im Plenum

Wir werden die Erkenntnisse und Ergebnisse von Arbeitsblatt 6 gemeinsam im Plenum besprechen.



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz

Nutzung des Python Packages "dtw"¹.

Autorin: Stephanie Hofmann

1 T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *J. Stat. Soft.*, doi:10.18637/jss.v031.i07

B.7 Präsentationsfolien

Auf den nachfolgenden Seiten sind die Präsentationsfolien abgedruckt. Alle Fragen, die an die Lernenden gestellt werden, sind mit einem Symbol gekennzeichnet, welches zwei Personen und eine Sprechblase mit einem Fragezeichen darstellt.



Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat
Einführung in die Problemstellung

Was ist Spracherkennung?



Sprachsignal Text

 CAMMP Workshop | Spracherkennung | 2

Anwendungsgebiete der Spracherkennung

Bei welchen Anwendungen wird Spracherkennung verwendet?



 CAMMP Workshop | Spracherkennung | 3

Anwendungsgebiete der Spracherkennung

Bei welchen Anwendungen wird Spracherkennung verwendet?

Diktat	Bedienung von Geräten ohne Hände
Bedienung über das Telefon	Sprachübersetzung
natürlichsprachliche Interaktion	...



 CAMMP Workshop | Spracherkennung | 4

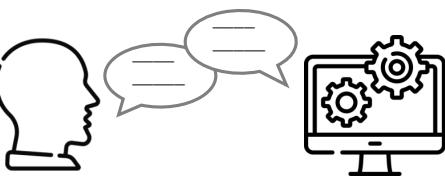
Spracherkennungssystem als typisches KI-System



Was verbindet ihr mit dem Begriff KI?
Welche Anwendungen der KI kennt ihr?

 CAMMP Workshop | Spracherkennung | 5

Spracherkennungssystem als typisches KI-System



 CAMMP Workshop | Spracherkennung | 6

Spracherkennungssystem als typisches KI-System

Timeline diagram showing the evolution of AI:

- 1956: Geburtsstunde der KI (Birth of AI)
- 2010: KI-Boom (AI Boom)
- 2011: Einführung von Siri (Introduction of Siri)

Grundprinzip eines Spracherkennungssystems

Diagram illustrating the principle of a speech recognition system:

- Sprachsignale mit bekannter Wortbedeutung: „one“, „two“, „three“, „four“
- Sprachsignal mit unbekannter Wortbedeutung: ?

Modellierungskreislauf

```

graph TD
    A[Reales Problem  
gesprochene Sprache erkennen] -- vereinfachen --> B[Vereinfachtes Problem]
    B -- mathematisch beschreiben --> C[Mathematisches Modell]
    C -- berechnen --> D[Mathematische Lösung]
    D -- interpretieren --> A
  
```

Wie ist ein Sprachsignal aufgebaut?

Graph of a digital speech signal:

- Y-axis: Intensität (Intensity)
- X-axis: Zeit in Sekunden (Time in seconds)
- Range: -600 to 400

digitales Sprachsignal

Wie ist ein Audiosignal aufgebaut?

Graphs illustrating the decomposition of an audio signal:

- Left graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows a complex waveform consisting of multiple frequencies.
- Right graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows a single pure tone (fundamental frequency).

Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne

Graphs illustrating the superposition of partial tones:

- Left graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the 1. Partial tone (fundamental frequency).
- Right graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the superposition of the 1. and 2. Partial tones.

Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne

Graphs illustrating the superposition of partial tones:

- Left graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the 1. and 2. Partial tones.
- Right graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the superposition of the 1., 2., and 3. Partial tones.

Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne

Graphs illustrating the superposition of partial tones:

- Left graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the 1. and 2. Partial tones.
- Right graph: Intensity vs. Time in seconds (0.000 to 0.100). It shows the superposition of the 1., 2., and 3. Partial tones.

Wie ist ein Audiosignal aufgebaut?

Überlagerung der Teiltöne

alle Teiltöne
ergibt das Audiosignal

Wie können wir die Teiltöne mathematisch beschreiben?

?

Wie können wir die Teiltöne mathematisch beschreiben?

- Mathematisch gesehen: Teiltöne sind nichts anderes als eine periodische Sinus oder Cosinus Schwingung
- Die Stimmablage erzeugt z.B. einen fast perfekten Sinuston:

Was sind charakteristische Eigenschaften eines Teiltons?

?

Was sind charakteristische Eigenschaften eines Teiltons?

- Lautstärke
- Tonhöhe
- Dauer

Einen Teilton mathematisch modellieren

$$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$$

Einen Teilton mathematisch modellieren

$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an

Einen Teilton mathematisch modellieren

$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an

Einen Teilton mathematisch modellieren

$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an
- f ist die Frequenz, gibt die Tonhöhe an und kann indirekt über die Periode $T = \frac{1}{f}$ bestimmt werden

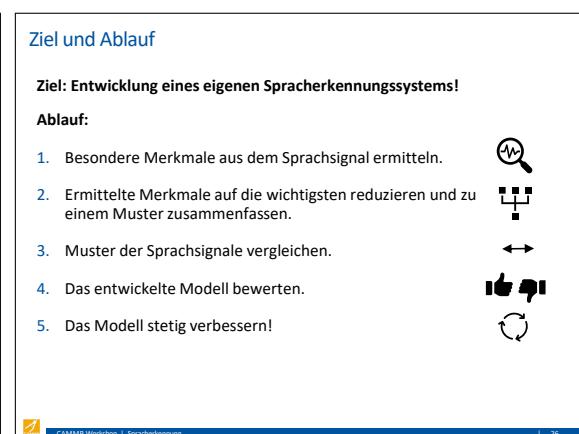
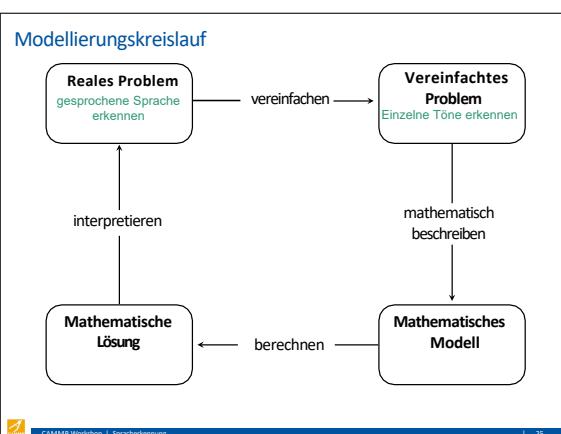
CAMMP Workshop | Spracherkennung | 23

Einen Teilton mathematisch modellieren

$g(t) = A \cdot \sin(f \cdot 2 \cdot \pi \cdot t)$

- A ist die Amplitude und gibt die maximale Auslenkung bzw. die Lautstärke des Tons an
- f ist die Frequenz, gibt die Tonhöhe an und kann indirekt über die Periode $T = \frac{1}{f}$ bestimmt werden

CAMMP Workshop | Spracherkennung | 24



Jetzt seid ihr dran ...

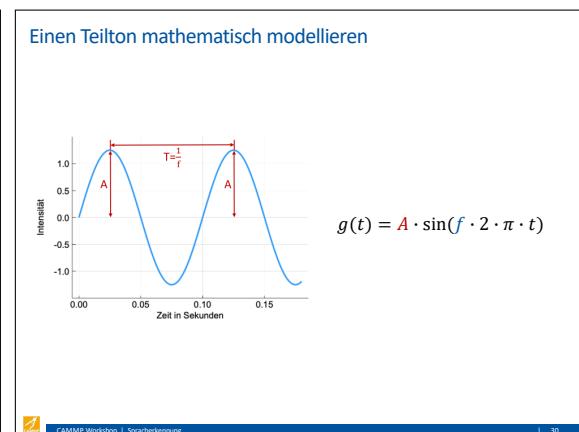
- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuer:innen!

CAMMP Workshop | Spracherkennung | 27

- Schritte zum Arbeitsmaterial
1. Öffne workshops.cammp.online
 2. Auf „Zugriff auf Lernmaterial“ und dann auf „Registrieren!“ klicken
 3. Account erstellen: der Username muss Präfix **cammp_** enthalten (z. B. **cammp_laura1234**)
 4. Auf „Anmelden!“ klicken, Accountdaten eingeben und einloggen
 5. Öffne die Datei **Willkommen_CAMMP**
 6. Im Dropdown Menü **Spracherkennung** auswählen und herunterladen
 7. Ordner **spracherkennung** und dann Ordner **worksheets** öffnen
 8. Los geht's mit Arbeitsblatt 1!
- CAMMP Workshop | Spracherkennung | 28

Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat

Besprechung AB1



Wie erhalten wir die einzelnen Teiltöne eines Tongemisches?

Fourieranalyse

Satz von Fourier
Jede periodische Funktion lässt sich als Summe von Sinus- und Cosinusfunktionen darstellen.

Joseph Fourier (1768 – 1830)

Fourieranalyse

Fourieranalyse

Jetzt seid ihr dran ...

• Bearbeitet die Arbeitsblätter!
• Aufgabenstellung sorgfältig lesen!
• Teamwork!
• Nutzt die Tipps!
• Nutzt das Internet!
• Fragt die Betreuer:innen!

Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat

Besprechung AB2

Fourieranalyse

Fourieranalyse

Modellierungskreislauf

Dreiklang vs. Sprachsignal

Dreiklang

Sprachsignal der Äußerung „two“

CAMMP Workshop | Spracherkennung | 39

Modellierungskreislauf

```

    graph TD
        A[Reales Problem  
gesprochene Sprache erkennen] -- vereinfachen --> B[Vereinfachtes Problem  
Einzelworterkennung]
        B -- mathematisch beschreiben --> C[Mathematisches Modell]
        C -- berechnen --> D[Mathematische Lösung]
        D -- interpretieren --> A
    
```

Reales Problem
gesprochene Sprache erkennen

Vereinfachtes Problem
Einzelworterkennung

Mathematische Lösung

Mathematisches Modell

interpretieren

mathematisch beschreiben

berechnen

CAMMP Workshop | Spracherkennung | 40

Jetzt seid ihr dran ...

Keine Panik!

ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuer:innen!

CAMMP Workshop | Spracherkennung | 41

Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat

Besprechung AB3

CAMMP

Ziel und Ablauf

Ziel: Entwicklung eines eigenen Spracherkennungssystems!

Ablauf:

1. Besondere Merkmale aus dem Sprachsignal ermitteln.
2. Ermittelte Merkmale auf die wichtigsten reduzieren und zu einem Muster zusammenfassen.
3. Muster der Sprachsignale vergleichen.
4. Das entwickelte Modell bewerten.
5. Das Modell stetig verbessern!

CAMMP Workshop | Spracherkennung | 42

Ziel und Ablauf

Ziel: Entwicklung eines eigenen Spracherkennungssystems!

Ablauf:

1. Besondere Merkmale aus dem Sprachsignal ermitteln.
2. Ermittelte Merkmale auf die wichtigsten reduzieren und zu einem Muster zusammenfassen.
3. Muster der Sprachsignale vergleichen.
4. Das entwickelte Modell bewerten.
5. Das Modell stetig verbessern!

CAMMP Workshop | Spracherkennung | 43

Ziel und Ablauf

Ziel: Entwicklung eines eigenen Spracherkennungssystems!

Ablauf:

1. Besondere Merkmale aus dem Sprachsignal ermitteln.
2. Ermittelte Merkmale auf die wichtigsten reduzieren und zu einem Muster zusammenfassen.
3. Muster der Sprachsignale vergleichen.
4. Das entwickelte Modell bewerten.
5. Das Modell stetig verbessern!

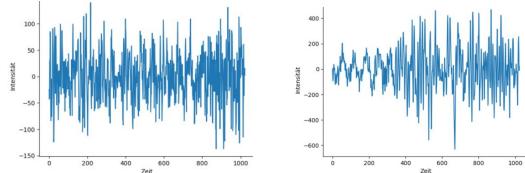
CAMMP Workshop | Spracherkennung | 45

Weshalb ist Spracherkennung so schwer?

CAMMP Workshop | Spracherkennung | 46

Weshalb ist Spracherkennung so schwer?

Gegeben: zwei **reale** Sprachsignale



Äußerung „two“ gesprochen von Person A Äußerung „two“ gesprochen von Person B

 CAMMP Workshop | Spracherkennung | 47

Sprachsignale klassifizieren

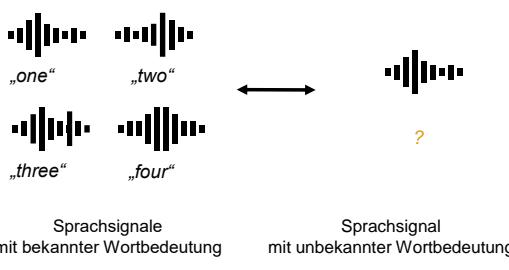
Regelbasierte Algorithmen

Feste Regeln

z.B. Wenn das Signal Merkmale A, B und C aufweist, dann hat es die Wortbedeutung „two“
 → da Sprache sehr variabel ist würde das nicht funktionieren

 CAMMP Workshop | Spracherkennung | 48

Sprachsignale klassifizieren



 CAMMP Workshop | Spracherkennung | 49

Sprachsignale klassifizieren

Regelbasierte Algorithmen

Feste Regeln

z.B. Wenn das Signal Merkmale A, B und C aufweist, dann hat es die Wortbedeutung „two“
 → da Sprache sehr variabel ist würde das nicht funktionieren

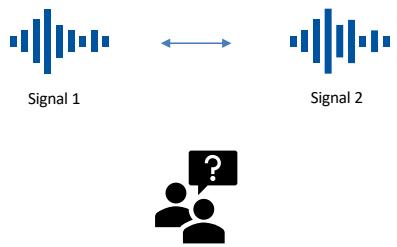
Maschinelles Lernen

Lernen von Daten

z.B. Vergleich des Signals mit Signalen deren Wortbedeutung bereits bekannt ist

 CAMMP Workshop | Spracherkennung | 50

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?



 CAMMP Workshop | Spracherkennung | 51

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?

- Zeitschritt für Zeitschritt vergleichen
- Ähnlichkeit messen:
 - Absolute Differenzen
 - Winkel zwischen Vektoren
 - Abstand zwischen Vektoren

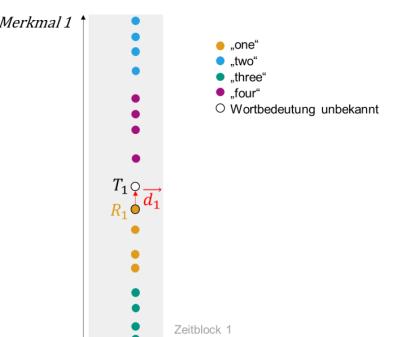
 CAMMP Workshop | Spracherkennung | 52

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?



 CAMMP Workshop | Spracherkennung | 53

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?



 CAMMP Workshop | Spracherkennung | 54

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?

Zeitblock 1

Merkmalsraum: Merkmal 2 (y-axis), Merkmal 1 (x-axis)

Legende: „one“ (orange), „two“ (blue), „three“ (green), „four“ (purple), „Wortbedeutung unbekannt“ (white)

Wie können wir die Ähnlichkeit zweier Sprachsignale messen?

Zeitblock 1

Merkmalsraum: Merkmal 2 (y-axis), Merkmal 1 (x-axis)

Legende: „one“ (orange), „two“ (blue), „three“ (green), „four“ (purple), „Wortbedeutung unbekannt“ (white)

Sprachsignale klassifizieren

Abstand berechnen

klassifizieren

„one“ „two“

„three“ „four“

Sprachsignale mit bekannter Wortbedeutung Sprachsignal mit unbekannter Wortbedeutung

erkanntes Wort

Modellierungskreislauf

Reales Problem: gesprochene Sprache erkennen

Vereinfachtes Problem: Einzelworterkennung

interpretieren

mathematisch beschreiben

Mathematische Lösung

berechnen

Mathematisches Modell: Klassifizierung mit Mustervergleich

Jetzt seid ihr dran ...

Keine Panik!

ORGANISIEREN!

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuer:innen!

Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat

Besprechung AB4

CC BY SA

A

Modellierungskreislauf

Reales Problem: gesprochene Sprache erkennen

Vereinfachtes Problem: Einzelworterkennung

interpretieren

mathematisch beschreiben

Mathematische Lösung: Vorhergesagte Klassenzugehörigkeit

berechnen

Mathematisches Modell: Klassifizierung mit Mustervergleich

Modellierungskreislauf

Reales Problem: gesprochene Sprache erkennen

Vereinfachtes Problem: Einzelworterkennung

interpretieren

mathematisch beschreiben

Mathematische Lösung: Vorhergesagte Klassenzugehörigkeit

berechnen

Mathematisches Modell: Klassifizierung mit Mustervergleich

Bewertungsideen



Bewertung

Ideen für Qualitätsmaße



Jetzt seid ihr dran ...

- Bearbeitet die Arbeitsblätter!
- Aufgabenstellung sorgfältig lesen!
- Teamwork!
- Nutzt die Tipps!
- Nutzt das Internet!
- Fragt die Betreuer:innen!



Alexa, Siri und co.: Spracherkennung und was das mit Mathe zu tun hat

Besprechung AB5

Trainings- und Testdaten

Wahrheitstabelle interpretieren

- Wie viele Signale wurden insgesamt richtig bzw. falsch klassifiziert?
- Welches Wort wurde am häufigsten bzw. seltensten richtig erkannt?
- Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

Wahrheitstabelle

		erkanntes Wort			
		one	two	three	four
tatsächliches Wort	one	6	2	0	3
	two	2	6	3	2
	three	0	2	5	1
	four	1	1	1	13



Wahrheitstabelle interpretieren

- Wie viele Signale wurden insgesamt richtig bzw. falsch klassifiziert?
- Welches Wort wurde am häufigsten bzw. seltensten richtig erkannt?
- Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

Wahrheitstabelle

		erkanntes Wort			
		one	two	three	four
tatsächliches Wort	one	6	2	0	3
	two	2	6	3	2
	three	0	2	5	1
	four	1	1	1	13



Wahrheitstabelle interpretieren

- Wie viele Signale wurden insgesamt richtig bzw. falsch klassifiziert?
- Welches Wort wurde am häufigsten bzw. seltensten richtig erkannt?
- Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

		Wahrheitstabelle				
		one	two	three	four	
tatsächliches Wort	one	6	2	0	3	
	two	2	6	3	2	
	three	0	2	5	1	
	four	1	1	1	13	
		one	two	three	four	
		erkanntes Wort	one	two	three	four
one	6	2	0	3		
two	2	6	3	2		
three	0	2	5	1		
four	1	1	1	13		



CAMMP Workshop | Spracherkennung | 71

Wahrheitstabelle interpretieren

- Wie viele Signale wurden insgesamt richtig bzw. falsch klassifiziert?
- Welches Wort wurde am häufigsten bzw. seltensten richtig erkannt?
- Bei welchen beiden Wörtern liegen die meisten Überschneidungen vor?

		Wahrheitstabelle				
		one	two	three	four	
tatsächliches Wort	one	6	2	0	3	
	two	2	6	3	2	
	three	0	2	5	1	
	four	1	1	1	13	
		one	two	three	four	
		erkanntes Wort	one	two	three	four
one	6	2	0	3		
two	2	6	3	2		
three	0	2	5	1		
four	1	1	1	13		



CAMMP Workshop | Spracherkennung | 72

Verschiedene Qualitätsmaße

- Was bedeutet es, wenn der Wert der Präzision nahe 0 bzw. nahe 1 liegt? Bei welchen Wörtern ist dies der Fall?
- Sind die Ergebnisse der Klassifikation zufriedenstellend? Falls nein, wo müssten die Ergebnisse noch verbessert werden?



CAMMP Workshop | Spracherkennung | 73

Aussagekraft der Qualitätsmaße

- Welches Spracherkennungssystem würdest du wählen?
- Was bedeutet dies für die Aussagekraft der Qualitätsmaße?

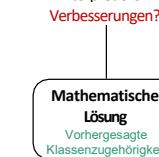


CAMMP Workshop | Spracherkennung | 74

Modellierungskreislauf

```

    graph TD
      A[Reales Problem  
gesprochene Sprache erkennen] -- vereinfachen --> B[Vereinfachtes Problem  
Einzelworterkennung]
      B -- mathematisch beschreiben --> C[Mathematisches Modell  
Klassifizierung mit Mustervergleich]
      C -- berechnen --> D[Mathematische Lösung  
Vorhergesagte Klassenzugehörigkeit]
      D -- interpretieren Verbesserungen? --> A
  
```



CAMMP Workshop | Spracherkennung | 75

Verbesserungsideen



CAMMP Workshop | Spracherkennung | 76

Verbesserungsideen

- mehr Daten
- die k-ähnlichsten Referenzmuster berücksichtigen
- Unterschiede im zeitlichen Verlauf berücksichtigen
- Bessere Qualität der Audioaufnahme (weniger Hintergrundgeräusche, bessere Mikros)
- ...



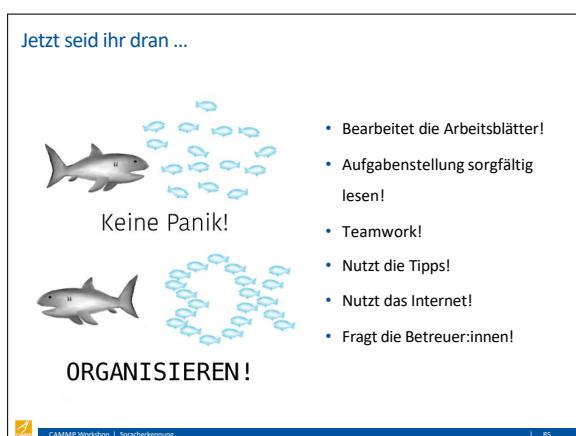
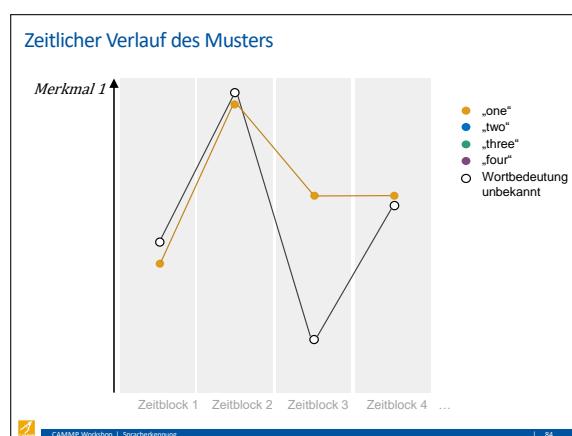
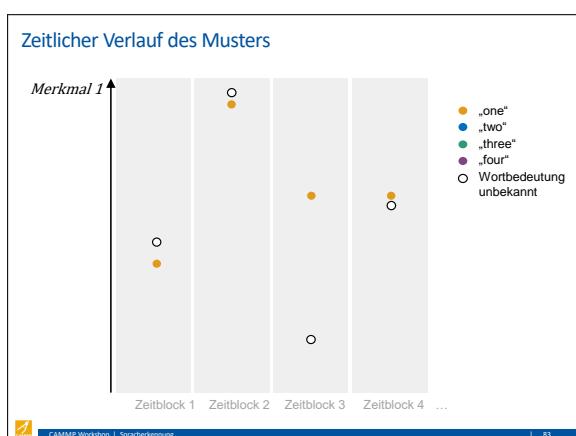
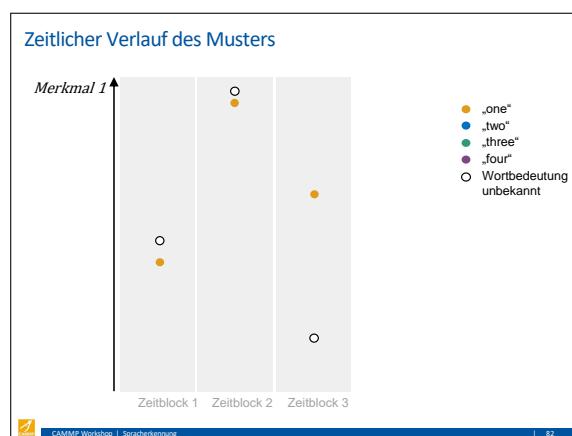
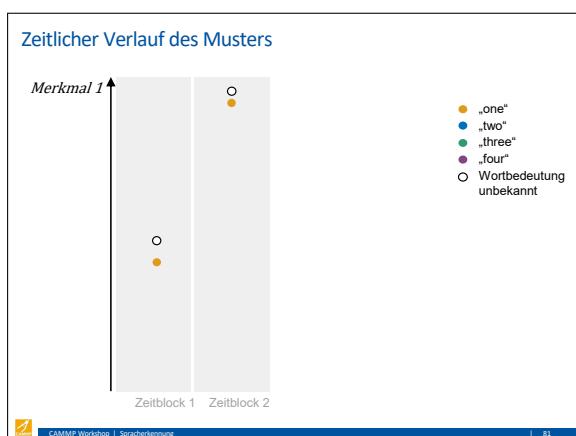
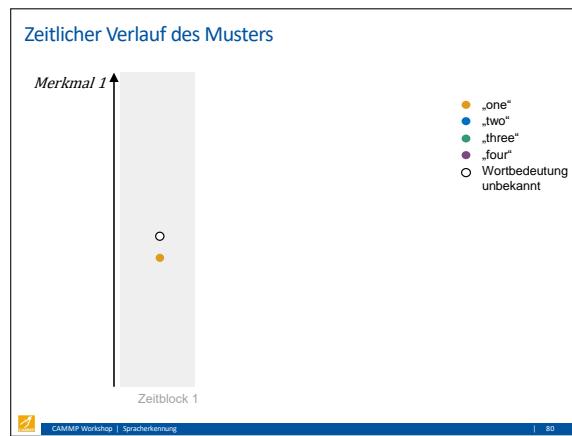
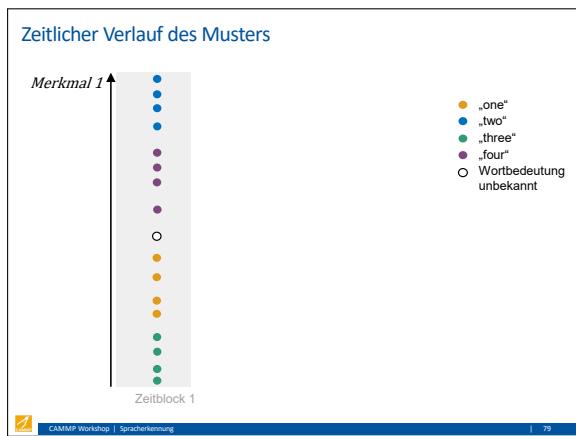
CAMMP Workshop | Spracherkennung | 77

Verbesserungsideen

- mehr Daten
- die k-ähnlichsten Referenzmuster berücksichtigen
- Unterschiede im zeitlichen Verlauf berücksichtigen
- Bessere Qualität der Audioaufnahme (weniger Hintergrundgeräusche, bessere Mikros)
- ...



CAMMP Workshop | Spracherkennung | 78



Dynamische Zeitanpassung

Diagram illustrating dynamic time adaptation. A 5x5 grid of numbers (3, 6, 5, 5, 1; 3, 5, 3, 3, 1; 1, 4, 1, 1, 3; 3, 1, 3, 5, 5; E/F, 4, 1, 1, 3) is shown with arrows indicating transitions between states. The grid is labeled with 'E/F' at the bottom left.

CAMMP Workshop | Spracherkennung | 87

Modellierungskreislauf

Flowchart illustrating the modeling cycle (Modellierungskreislauf):

- Reales Problem** (gesprochene Sprache erkennen) → **Vereinfachtes Problem** (Einzelworterkennung)
- Vereinfachtes Problem** → **Mathematisches Modell** (Klassifizierung mit Mustervergleich und DTV)
- Mathematisches Modell** → **Mathematische Lösung** (Vorhergesagte Klassenzugehörigkeit)
- Mathematische Lösung** → **Reales Problem**

Diagram illustrating the modeling cycle with a graph showing 'Zertisse von Nutzer B' (User B's buzzes) vs 'Zertisse von Nutzer A' (User A's buzzes) and a scatter plot of 'Zertisse von Nutzer B' vs 'Zertisse von Nutzer A'.

CAMMP Workshop | Spracherkennung | 88

Modellierungskreislauf

Flowchart illustrating the modeling cycle (Modellierungskreislauf):

- Reales Problem** (gesprochene Sprache erkennen) → **Vereinfachtes Problem** (Einzelworterkennung)
- Vereinfachtes Problem** → **Mathematisches Modell** (Klassifizierung mit Mustervergleich und DTV)
- Mathematisches Modell** → **Mathematische Lösung** (Vorhergesagte Klassenzugehörigkeit)
- Mathematische Lösung** → **Reales Problem**

Diagram illustrating the modeling cycle with a truth table (Wahrheitstabelle) for a logic gate:

		one	two	three	four
one	one	0	1	1	
two	0	11	1	1	
three	0	2	5	1	
four	0	0	0	16	

CAMMP Workshop | Spracherkennung | 89

Herausforderungen für unseren Spracherkenner

- Welche Probleme treten bei der Anwendung auf?
- Was können Gründe für diese Probleme sein?

CAMMP Workshop | Spracherkennung | 90

Herausforderungen für unseren Spracherkenner

- Welche Probleme treten bei der Anwendung auf?
- Was können Gründe für diese Probleme sein?

CAMMP Workshop | Spracherkennung | 91

Anwendungsgebiete für unseren Spracherkenner

Für welche Zwecke ist ein solcher Spracherkenner geeignet?

CAMMP Workshop | Spracherkennung | 92

Anwendungsgebiete für unseren Spracherkenner

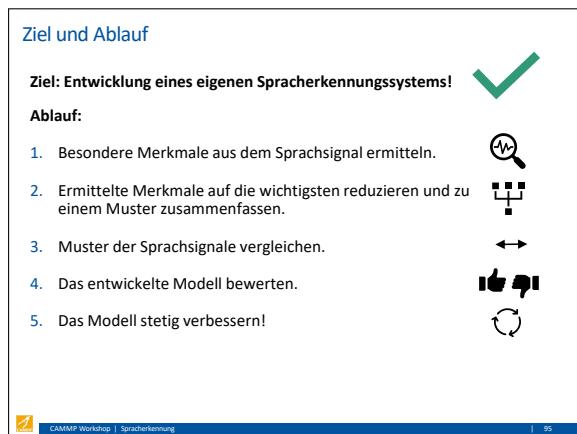
Für welche Zwecke ist ein solcher Spracherkenner geeignet?

CAMMP Workshop | Spracherkennung | 93

Kritische Reflexion von Spracherkennung

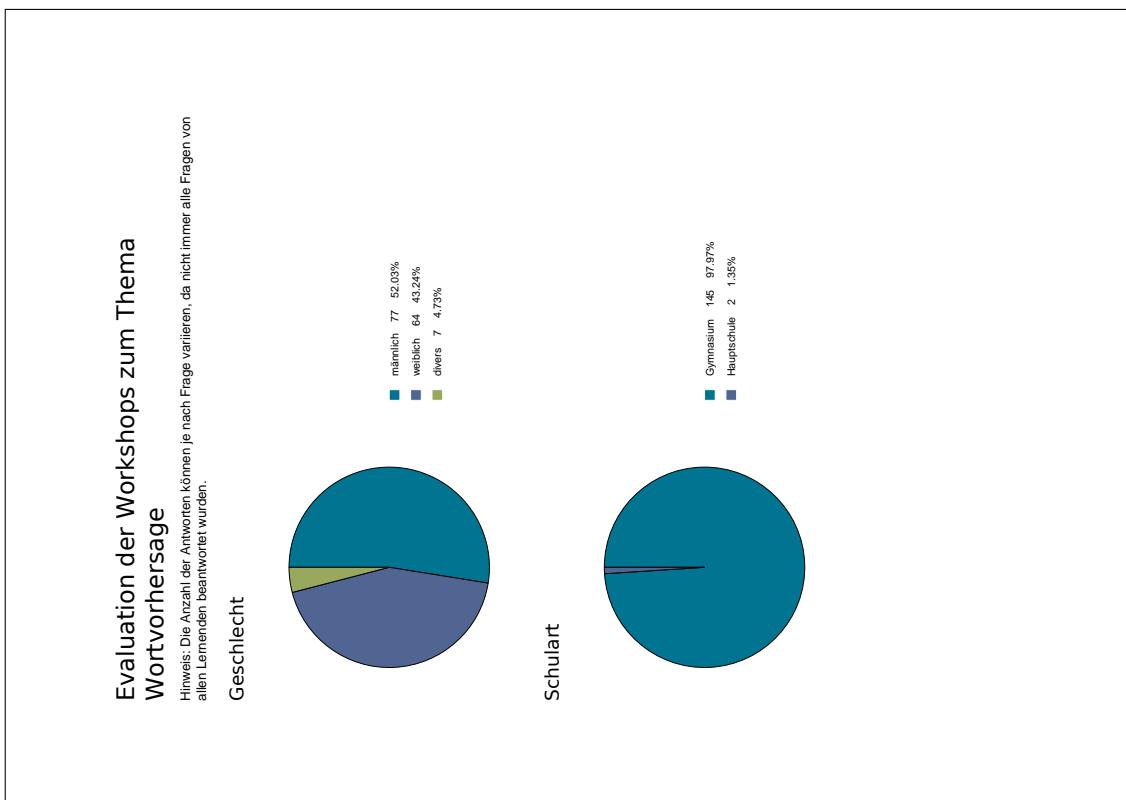
- Problematische / ethisch bedenkliche Anwendungen der Spracherkennung?
- Nützliche / sinnvolle Anwendungen der Spracherkennung?
- Problematisch/ ethisch bedenkliche Aspekte bei der Entwicklung von Spracherkennern?

CAMMP Workshop | Spracherkennung | 94

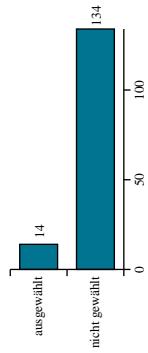


C Evaluationen der Durchführungen

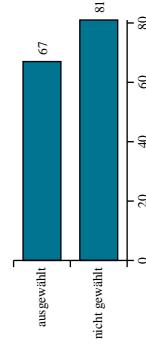
C.1 Evaluationen der Durchführungen zum Thema Wortvorhersage



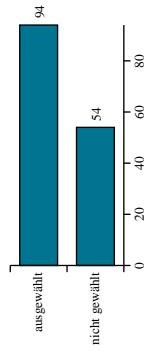
Leistungskurs: Informatik



Leistungskurs: andere



Leistungskurs: Mathematik



Leistungskurs: Physik

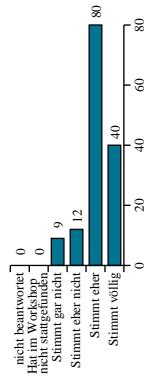


Leistungskurs: andere (offene Eingabe)
 Einträge bei der Frage "Leistungskurs: andere (offene Eingabe)" waren:

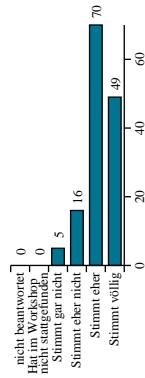
Chemie
 Politik
 Biologie bili, Wirtschaft
 Chemie
 Englisch
 Biologie, Musik
 Französisch
 Chemie, französisch
 Englisch
 Latein
 Englisch
 Informatik Basiskurs
 Chemie
 GMK
 English, Sport
 Deutsch, Bk, Biologie
 Bio, Bk, Deutsch
 Chemie
 Englisch
 Chemie, Kunst
 Wirtschaft, Biologie
 Englisch
 Musik
 Musik
 Chemie
 gewählt: Deutsch, Französisch, Chemie
 Geschichte, Englisch und Deutsch
 Deutsch
 Chemie, Wirtschaft
 Wirtschaft, Englisch

Wirtschaft
 deutsch, gemeinschaftskunde, englisch
 Deutsch, Kunst, Englisch
 Deutsch Englisch Kunst
 Chemie
 Englisch, Bio und Geographie
 Latein
 erdkunde deutsch
 Erdkunde
 Englisch
 Sozialkunde
 Deutsch, Chemie
 Chemie Erdkunde
 Chemie, Erdkunde
 Englisch
 Erdkunde
 Erdkunde
 Deutsch, Erdkunde
 Englisch, Geographie
 Deutsch, Kunst, englisch
 Deutsch, Bio, Geo
 Deutsch, Englisch, Sport
 Englisch, Geographie, Deutsch
 English, Biologie, Geographie
 Kunst, Deutsch, Englisch
 Deutsch, BK, Bio
 englisch, chemie, geographie
 Chemie, Englisch
 Chemie
 Bio, Englisch
 Chemie, Englisch

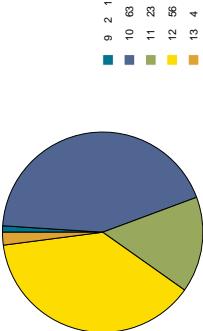
Bewertung des Workshops: Durch den Workshop habe ich mathematisches Modellieren besser begreifen.



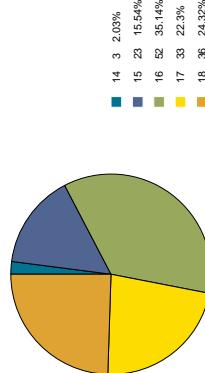
Bewertung des Workshops: Der Vortrag über mathematische Modellierung war hilfreich.



Jahrgangsstufe

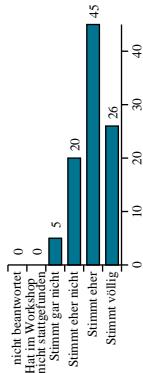


Alter

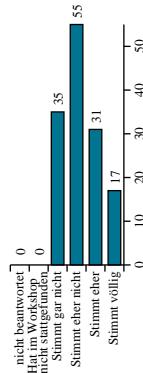


C.1 Evaluationen der Durchführungen zum Thema Wortvorhersage

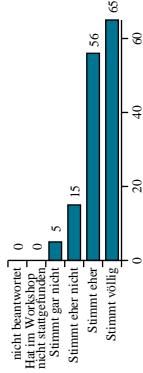
Bewertung des Workshops: Die Einführung in Jupyter Notebooks und Julia war hilfreich.



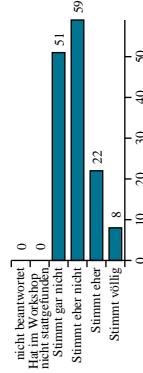
Bewertung des Workshops: Der Umgang mit der Programmiersprache Julia fiel mir schwer.



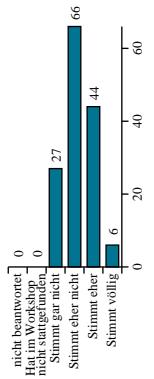
Bewertung des Workshops: Die Gestaltung der Arbeitsblätter fand ich gut.



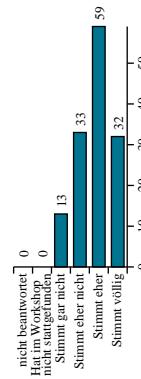
Bewertung des Workshops: Die Aufgaben waren zu einfach.



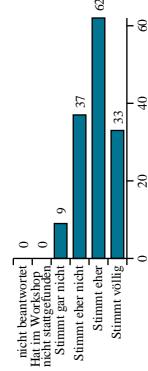
Bewertung des Workshops: Die Aufgaben waren zu schwierig.



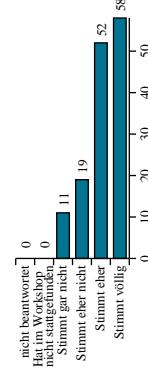
Bewertung des Workshops: Die Tipps / Hilfekarten waren hilfreich.



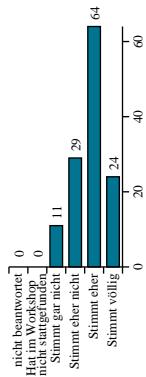
Bewertung des Workshops: Ich konnte die Aufgaben eigenständig bearbeiten.



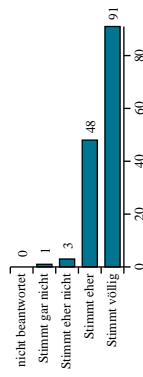
Bewertung des Workshops: Ich fand die automatischen Rückmeldungen, die ich nach der Eingabe meiner Lösungen im Code erhalten habe, hilfreich.



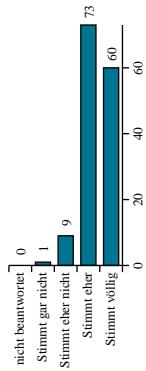
Bewertung des Workshops: Die Zusatzaufgaben haben mir sehr gefallen.



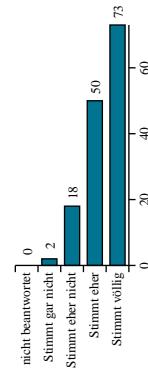
Weiterführende Fragen: Die Lern- und Arbeitsatmosphäre war angenehm.



Weiterführende Fragen: Die Inhalte wurden klar vermittelt.



Weiterführende Fragen: Die Lern- und Arbeitszeiten waren angemessen (nicht zu lang oder zu kurz, mit ausreichenden Pausen).



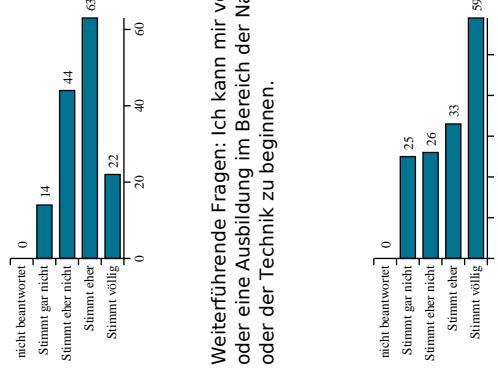
Weiterführende Fragen: Das Projekt hat mein Interesse an Themen der Naturwissenschaften und Technik gesteigert.



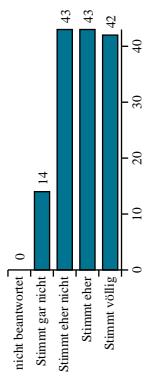
Weiterführende Fragen: Durch den Workshop habe ich interessante Berufs- und Studienmöglichkeiten kennengelernt.



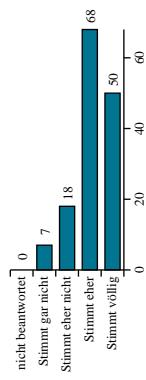
Weiterführende Fragen: Ich habe in diesem Workshop viel Neues gelernt, was mir für die Schule, für ein Studium und / oder Beruf weiterhelfen kann.



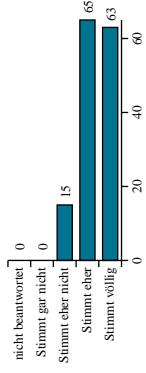
Weiterführende Fragen: Ich würde so einen Workshop gerne noch einmal besuchen.



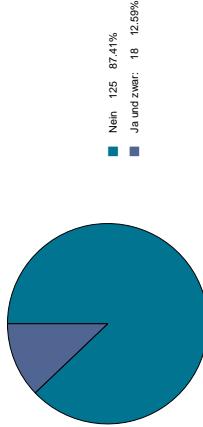
Weiterführende Fragen: Ich würde so einen Workshop auch anderen weiterempfehlen.



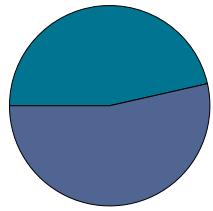
Weiterführende Fragen: Alles in allem hat mir der Workshop gut gefallen.



Nicht gefallen



Besonders gefallen

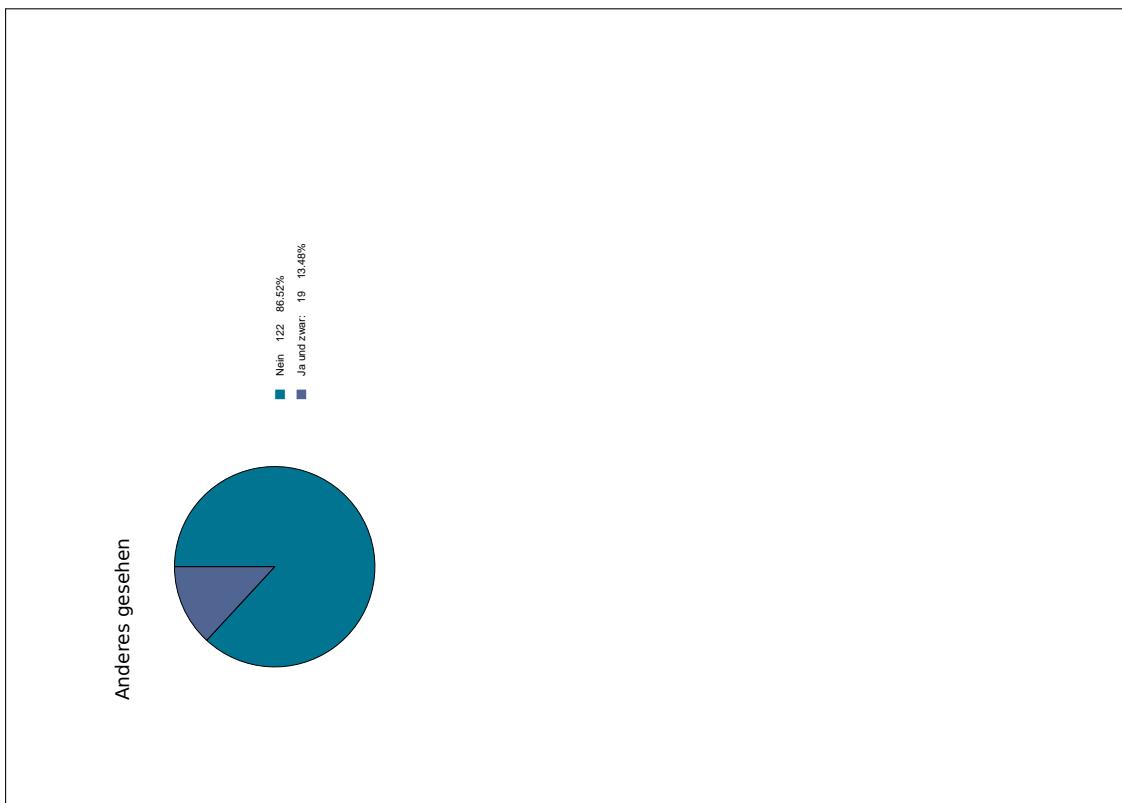


Nein 64 46,04%
Ja und zwar: 75 53,96%

- Nicht gefallen: Ja und zwar
- Einträge bei der Frage "Nicht gefallen: Ja und zwar" waren:
- Ich hätte gerne irgendwo die Musterlösungen eingesehen.
 - Es war nicht sehr schlimm, aber es gab schon einen gewissen Zeitdruck zumindest am ersten Tag, der mich als nicht ganz so schnell arbeitende Person, etwas gestresst hat
 - immer nur in der gleichen Gruppe zu arbeiten, gerne etwas durchmischen
 - Die Beschreibung bei auf der MINT EC Seite war sehr stark abweichend, zu dem, was dann tatsächlich gemacht wurde
 - Es gab keine Einführung in die Programmiersprache und die Gruppeneinteilung war nicht so gut
 - Meine Gruppe hat nicht mit mir gearbeitet
 - Am Anfang war die Atmosphäre etwas angespannt, am Ende aber nicht mehr...
 - Zu wenig Awechselung
 - Zu schwierig
 - Code hat teilweise sehr lange zum Ausführen gebraucht
 - Manchmal war man bei den Aufgaben zu überfordert und dann hatte man keine Lust mehr
 - Die Benutzung des Laptops wäre durch eine Computermaus sehr viel einfacher
 - Zu wenig eigenständige, kreative Arbeit
 - Aufgaben sehr kompliziert gestellt und bessere Einführung wäre nötig
 - Die Grundsprache der Programmierung war mir unklar am Anfang, wo man Anführungszeichen oder Klammern benutzt
 - Es war öfters der Fall, daß alles zu schnell ging und die Aufgaben dadurch schwerer zu bearbeiten waren

Besonders gefallen: Ja und zwar
 Einträge bei der Frage „Besonders gefallen: Ja und zwar“ waren:
 Die Gestaltung der Arbeitsblätter und die kleine Gruppengröße und
 somit bessere Zusammenarbeit :)
 sehr gute Vorbereitung der Materialien, strukturierte Vorgehensweise,
 angemessene Pausen, Freundlichkeit der Kursleiterinnen
 die Leute (die "Lehrer" sowie auch die anderen Schüler)
Aufgaben & Organisation
 Gerade die Arbeitsblätter waren sehr gut gestaltet, und auch die
 Besprechungen waren anders als bei dem Solarstromer Workshop
 zeitlich besser getaktet und es wurde darauf mehr geachtet, dass alle
 Teilnehmer wirklich mitkommen
 Verständliche Texte undfordernde, aber nicht unmögliche Aufgaben,
 haben das Verständnis des Themas sehr erleichtert
 die freien, problemorientierten Aufgaben & die Diskussionen im Plenum
 (mit Einbezug von Studien, ...)
 die Arbeitsblätter
 der Ablauf zwischen gemeinsamen Besprechungen, neuen Input und
 Gruppenarbeiten
 die Vorträge
 Der Service & die Nachbesprechungen, Wiederholungen
 Die Sprache Julia
 Die Möglichkeit am Ende die Aufgaben sehr frei zu bearbeiten
 Den Aufbau der Arbeitsblätter
 Hilfreiche, nette Betreuer
 Die schnelle Erreichbarkeit der Betreuer/-innen
 Wenn man Hilfe brauchte kam immer sofort jemand,
 dass nichts zu eicht oder zu schwer war und man alles gut mit dem
 partner erarbeiten konnte
 die gute Strukturierung des Erarbeiten der neuen Inhalte
 Die sehr gut vorbereiteten Arbeitsblätter und die freundlichen
 Leiter/-innen
 Viel Selbstständiges Arbeiten
 selber Programmieren

Meine Fragen wurden ausführlich und exakt beantwortet.
 Hilfe von „Trainern“, Einblick in den Bereich Wortvorschläge in den
 Suchmaschinen
 Die Arbeit in den Gruppen.
 Ja, zu sehen, dass aus kleinen Schritten ein Programm
 zusammengesetzt wurde, dass einen in sich stimmigen Satz ausgeben
 konnte
 Abwechslung zwischen Arbeitsphase, Besprechung und die zwei Pausen
 dass wir mehr oder weniger alleine und selbstständig arbeiten durften.
 Man wurde in den Arbeitsblättern gut geleitet
 Dass wir uns immer nach Bearbeitung eines Arbeitsblattes besprochen
 haben
 Freiere Programmieraufgaben
 Dass man immer Fragen stellen konnte
 Kein Musik zu haben)
 man konnte sich vieles selbst erarbeiten
 Mit anderen interessierten Menschen zu arbeiten.
 die Pausen
 XXX und XXX haben alles prima erklärt, sodass ich ohne Probleme die
 Aufgaben lösen konnte.
 keine Schule
 Ich fand alles gut
 die Leiter des Kurses
 Aufbau
 -
 gutes Team der Referenten
 Mathematisches modellieren
 Die Gestaltung vom Workshop war sehr nice
 Einblick ins Programmieren
 Die Betreuer
 die Möglichkeit, mit den Arbeitsblättern relativ frei und im eigenen
 Tempo zu arbeiten
 Der Aufbau der Arbeitsblätter; Freies und selbstständiges Arbeiten



Etwas neues zu lernen
Mathe und so
war kras
Das eigenständige Arbeiten
Alles war super
WLAN
Thema war aktuell
Der Programmierteil
Das Arbeiten mit dem Laptop
Die Themen zu verstehen
Das programmieren
Die netten Studenten, die uns bei jedem Problem helfen konnten und auch die lehrreichen Präsentationen gehalten haben.
Das fast eigenständige Arbeiten
Zusammenarbeit
Der Vergleich mit dem Alltag und dass die Studenten geholfen haben
Den Einblick in das Thema bekommen
die Unterstützung
Die netten Stunden, die uns alles erklärt und auch immer zur Seite standen, wenn wir Hilfe gebraucht haben oder fragen hatten
viel
Das Ausprobieren der Lösungsweg in den Feldern
Das die Thematik an was alltägliches wie wortvorschläge angewandt wurde was es nahbarer und relevanter macht
Das selbst(er)arbeiten
Selbstständigkeit, aber wenn man eine Frage hatte waren die Betreuer hilfsbereit
Die Selbstständigkeit, aber auch, wie der Workshop generell aufgebaut war

Anderes gesehen: ja und zwar

Einträge bei der Frage "Anderes gesehen: ja und zwar" waren:

- Fortsetzung mit komplexeren, aber auch genaueren Methoden/Möglichkeiten ein Wortvorhersagemodell zu erstellen (z.B. Einsatz Neuronaler Netze)
- Anwendung des Verfahrens des steilsten Abstiegs
- Empfehlung von weiterführenden Links
- mehr konkrete Beispiele zur mathematischen Modellierung der freundliche Umgang aller Tutoen mit uns :)
- eine kleine einführung fürs programmieren
- Programmierung (Einführung & Erklärung)
- Wie man eine KI programmiert.
- wie das Optimierungsverfahren in Ansätzen funktioniert
- Neuronale Netze
- ich hätte gerne unser Vorhersagemodell mal selbst ausprobiert.
- Neuronale Netzwerke
- wie groß wirkliche Trainingsdatensätze sind
- mehr zum Thema "machine learning" und Gurkensalat
- etwas mehr Praxis,Beispiele
- komplexere /detailliertere Beschäftigung mit dem Thema irgendwas
- Bestimmt, mir fällt gerade nichts Bestimmtes ein, aber mehr erfahren ist eigentlich immer schön

Lernzuwachs: [01]

Einträge bei der Frage "Lernzuwachs: [01]" waren:

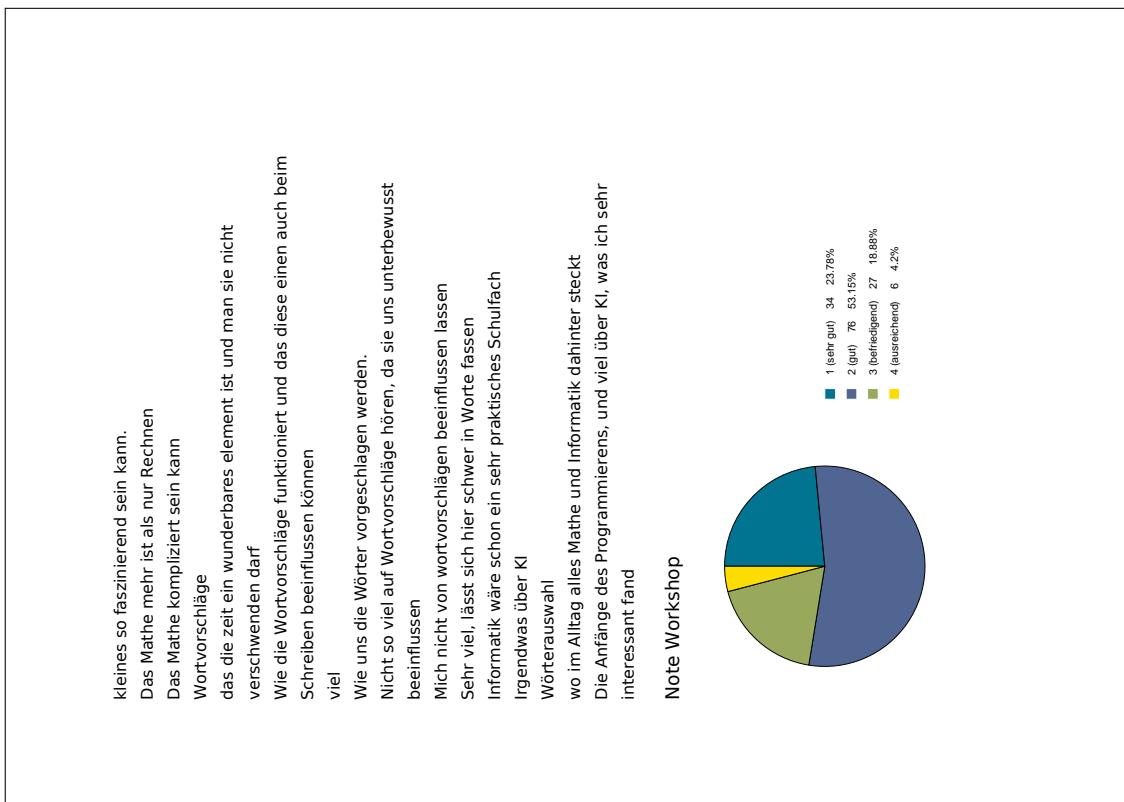
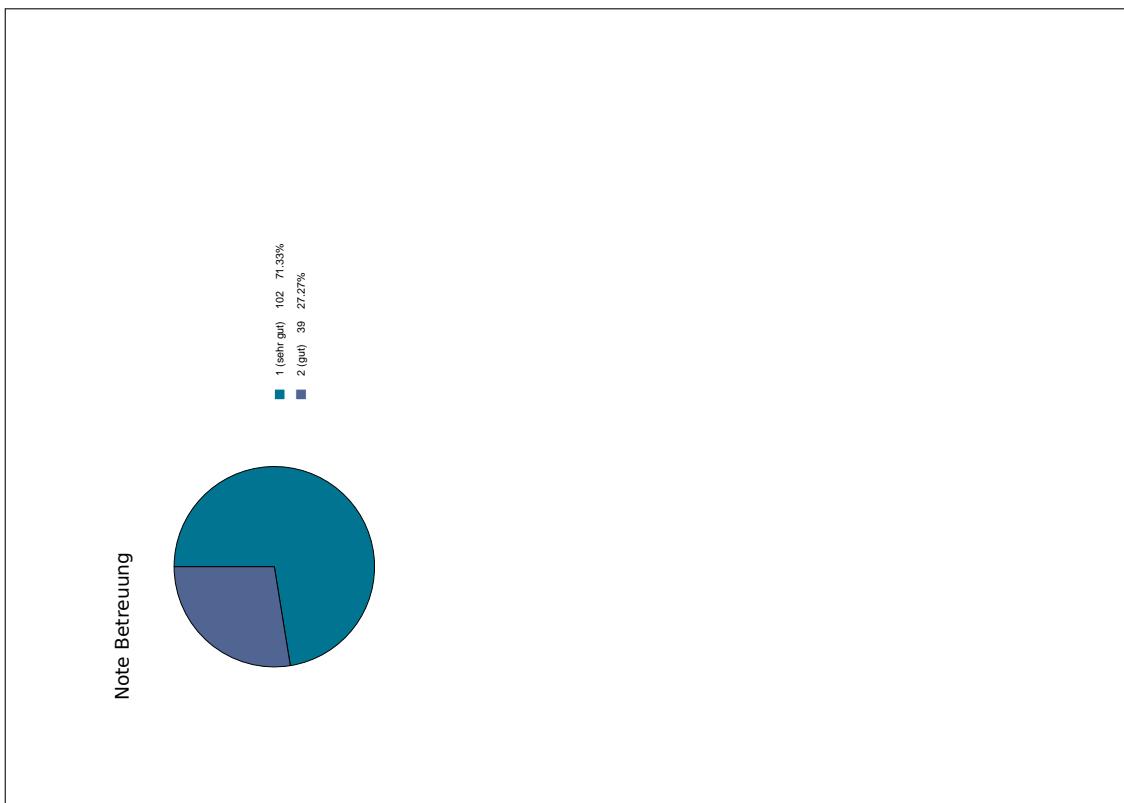
- Funktionsweise mathematischer Modellierung & Probleme hinter Wortvorschlägen zu lösen
- Einblick in das Prinzip der Wortvorschläge, Verknüpfung verschiedener Modelle mittels Wahrscheinlichkeiten
- Ich würde gerne nochmehr Workshops machen!
- mathematische Modellierung macht Spaß, Verfahren des steilsten Abstiegs; Gradienten
- Ich fand der Einblick in die Generierung von Wortvorschlägen sehr interessant und es war ein sehr gutes Beispiel dafür wie Mathematische Modellierung in Alltag Anwendung findet
- Funktionsprinzip von Wortvorschlägen, Weiteres Verständnis von mathematischer Modellierung
- viel über KI-Wortvorschläge und wie man diese programmiert
- mehr Programmierkenntnisse; Idee von mathematischer Modellierung bekommen
- Die Vorteile und Nachteile von KI's.
- Es muss nicht immer alles schnell gehen.
- Wie KI überhaupt funktioniert!
- Ein besseres Verständnis des Prozesses der mathematischen Modellierung
- Ungefähr wie Wortvorschläge funktionieren
- viele neue Begriffe in der Welt der Informatik, Inwiefern ein Informatikstudium zu mir passt
- Programmiergrundlagen, wie KIs für Wortvorschläge funktionieren und auf was für Dinge diese achten.
- mit Julia zu programmieren
- Programmierkenntnisse, wie Mathe auch in alltäglichen Sachen angewandt werden kann
- Erfahrungen im Modellieren und Programmieren
- Ich fand es interessant, zu lernen, dass es auch andere Machine-Learning-Methoden außer Neuronalen Netzen für NLP gibt.
- programmieren

Wortvorschläge sind komplexer als man denkt

- Wie die Modelle funktionieren und wie sie zusammen bzw. im Wechsel arbeiten.
- Die folgen die ständiges benutzen von Wortvorschlägen hat.
- Ausweitung von Listen, Nutzung der Daten
- Ich habe einen Einblick in das System der Wortvorschläge bekommen.
- andere Formen von Mathematik
- Es hat Spaß gemacht, ich werde nun meinen Vater zwingen, mir programmieren beizubringen.
- Wie Mathe in der Realität benutzt wird
- das Wahrscheinlichkeiten nicht so meins ist
- mathematisches modellieren
- Ich studiere kein mathe
- Mathe kann lustig sein
- wie wortvorschläge gemacht werden
- Wie manche Dinge so stark von der Mathematik abhängen
- Wie Mathematik mit der echten Welt verbunden wird, und wie man besser mit Programmen arbeitet
- Julia; Die Idee, den gewichteten Mittelwert mehrerer Modelle zu verwenden; Kreuzentropie
- Ich habe gelernt dass Java besser als Julia ist :)
- Kreuzentropie und Mathematischemodellierung
- Ich habe gelernt wie man Stochastik bei realen Problemen anwenden kann. Besonders interessant fand ich das Thema Kreuzentropie.
- Dass die Berechnung für die Wortvorschläge komplizierter ist als gedacht...
- Über die Programmiersprache.
- Funktionsweise der Vorschläge/das Prinzip dahinter
- Wie genau die Vorschlagsfunktion meiner Tastatur funktioniert und dass mein Handy schlauer ist als ich dachte.
- Ein besseres Verständnis von mathematischer Modellierung.

Einblick in Bereiche, welche später mal im Beruf/ Leben hilfreich sein können

- Mathematisches modellieren
- Ich habe gelernt, dass man eigentlich jedes Problem mathematisch modellieren und lösen kann.
- wie man an Probleme rangeht
- die Komplexität der Mathematik, man kann fast alle Probleme mithilfe von Mathematik bewältigen
- Mathematische modellierung
- Wie das Wortvervollständigen ungefähr funktioniert und wie Mathe im Alltag verwendet wird.
- Wie krass unpersonlich vorgegebene Antworten sind
- Durch den Workshop habe ich nun ein besseres Verständnis für die mathematische Modellierung.
- Kein Mathe studieren
- Wie schwierig es ist ein mathematisches problem in computersprache darzustellen
- mit Julia zu arbeiten
- war cool
- die Funktion von Wortvorschlägen
- Ganz schön viel
- Geduld und Konzentration bringen einen weiter
- Informatik
- Besser mit der Programmiersprache umzugehen
- Informatik
- nichts
- mehr zur Modellierung
- Wie man Mathematische Probleme durch Programmieren löst
- Wer lesen kann ist klar im Vorteil
- Das mathematische Modellieren
- Das ich mich mit diesem Thema nicht mehr beschäftigen werde
- Wie das System mit den Wortvorschläge funktioniert
- Was alles hinter den Wortvorschlägen steckt und dass doch so etwas



Abschließender Kommentar: [01]
Einträge bei der Frage "Abschließender Kommentar: [01]" waren:
Wirklich herausragende Freundlichkeit und Hilfsbereitschaft der
Kursleitenden
war toll!

Ich kann nur sagen, alles in allem ein gelungener Workshop, der mir
persönlich sehr gut gefallen hat. Alles war top organisiert und es wurde
sehr darauf geachtet, dass jeder bei dem Thema mitgenommen wird.

Großes Lob an dieser Stelle

Schwierigkeitsgrad der Texte und Aufgaben war sehr angenehm,
manchmal waren Aufgaben schwerer, da eine wichtige Information zu
Aufgabe aus dem letzten Kapitel kam, wobei die Lösungen dann
geholfen haben. Unterstützung und Kontrolle durch die "Mentoren" war
hilfreich. Vielen Dank :)

Ich bin sehr froh, dass ich an solch einer fantastischen Veranstaltung
teilnehmen durfte! Vielen Dank!

Ich fand es super, nur hätte ich mir irgendwo eine Musterlösung
gewünscht. Außerdem wäre es schön, wenn man euch irgendwie noch
erreichen könnte, falls man noch einmal Fragen zu den Themen hat.
vielen Dank für den tollen Workshop!

Eine meiner schönsten Erlebnisse!

Dadurch dass das Camp über Videokonferenz stattgefunden hat, musste
man fast durchgehend konzentriert sein, um alles zu schaffen und zu
verstehen. Deshalb war besonders der erste Tag sehr anstrengend.
Austausch innerhalb der gesamten Gruppe im digitalen Format
schwieriger, sehr nette Betreuer/-innen, wenige technische

Schwierigkeiten
Hat sehr viel Spaß gemacht, danke!

es war super
für AB5 und 7 wäre mehr Hilfestellung schön gewesen, ansonsten hat
mir alles sehr gut gefallen, die Informationstexte waren sehr
verständlich, sodass ich viel mitnehmen konnte. Danke!
Alles, in allem guter Workshop. Freitag nachmittags war nur einfach
wenig Konzentration und Motivation da. Gruppeneinteilung könnte

besser gewesen sein, und ich hätte mir eine Einführung in die
Programmiersprache gewünscht, vor allem, weil doch einiges anders als
zu Java (Lernen wir in der Schule in Inofmatik) ist.

Gute Einführung

Der Inhalt wurde gut präsentiert und erklärt und es war angenehm
zuhören.

Gute Trainer, guter Workshop insgesamt!

Ich fand alles in allem sehr gut gestaltet und alles hat gut aufeinander
aufgebaut. Obwohl ich noch nie Kontakt mit Programmieren hatte hab
ich das meiste gut hinbekommen.

Ein interessanter Einblick in ein Thema, was wir täglich verwenden!

Vielen Dank, es war sehr gut. Betreuer waren nett und hilfreich.

Es war interessant und hat Spaß gemacht

Mehr programmieren, lieber in Python anstatt Julia, es ist weiter
verbreitet
Die die es vorgestellt haben waren sehr nett
die aufgaben hätten simpler erklärt werden können aber sonst gut
Danke für den Workshop.

Ich würde eine umfangreichere Einführung ins Thema vorschlagen. Die
Aufgabenstellung war meistens etwas kompliziert und ein
bisschen schwierig.

Das Thema wurde gut und verständlich beigebracht und es hat generell
viel Spaß gemacht mit Mathe Probleme zu lösen :) Was man jedoch
verbessern könnte wären bessere Fehlermeldungen, wenn im Programm
(Syntax) nicht stimmt)

Es war schön

Mehr praktische programmierung wäre schön gewesen, aber auch so
war der Workshop hilfreich und cool. Alles in allem ein gelungener und
nice Tag!:

Ich fand euch ehrlich voll gut
sehr gut gemacht, weiter so! :)
gut strukturiert
hii
War nice

C.1 Evaluationen der Durchführungen zum Thema Wortvorhersage

Alles in allem war der Workshop aufschlussreich und interessant, zwischendurch hat es sich allerdings etwas gezogen, was vielleicht auch an der Luft des Seminarraumes lag.
Tolle Betreuer nur das Thema hat mir nicht so gefallen
-
Vielleicht ein bisschen mehr Pause an so warmen Tagen, sonst war alles gut.

habe selber kein großes Interesse am Thema, der Vortrag war aber an sich gut gestaltet und die BetreuerInnen waren sehr nett

Nächstes mal ein bisschen mehr pause
Sehr liebe Betreuer und sehr sauber

langweilige Gestaltung, dadurch sehr eintönig

Alles gut

Die Präsentation und die Arbeitsblätter waren gut gestaltet und uns wurde alles gut erklärt.

Nette Betreuer, künstliche Intelligenz zu schwer. Viel Glück noch im Studium!

Super ! Ich habe sehr sehr viel gelernt und bin sehr froh, dass ich hier sein durfte. Danke und ein großes Lob an die Studenten, ihr habt das echt super gemacht.

ich seid alle sehr nett

Alles in allem ein sehr schöner und lehrreicher Tag, nette Leute und Interesse weckende Beiträge

war sehr schön, nur weiter so :)

Es war eine sehr gute Vorstellung und die Aufgaben haben mir gut gefallen{thumbsup_tone1}

Sehr gute und schnelle Hilfestellung bei aufkommenden Problemen

Es war ein interessanter Workshop

Vielen Dank für den tollen Workshop ! An sich fand kch alles ziemlich gut!

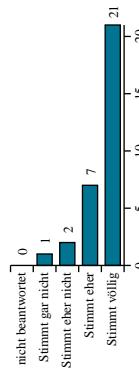
Insgesamt sehr nett alles :)

Nö

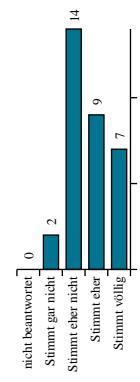
Vielen Dank

sehr guter Workshop
Vielen Dank für diesen Workshop, er ist sehr hilfreich und interessant, und wie ihr die Themen erklärt und wie die Arbeitsblätter aufgebaut sind, ist sehr toll.

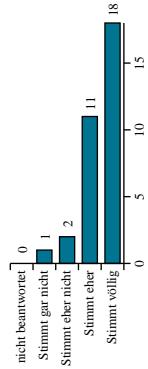
offene Aufgaben: Die offenen Aufgaben haben mir gefallen.



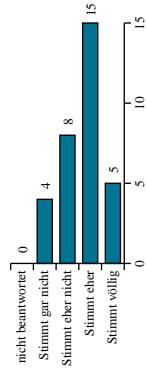
offene Aufgaben: Die offenen Aufgaben waren für mich schwierig zu lösen.



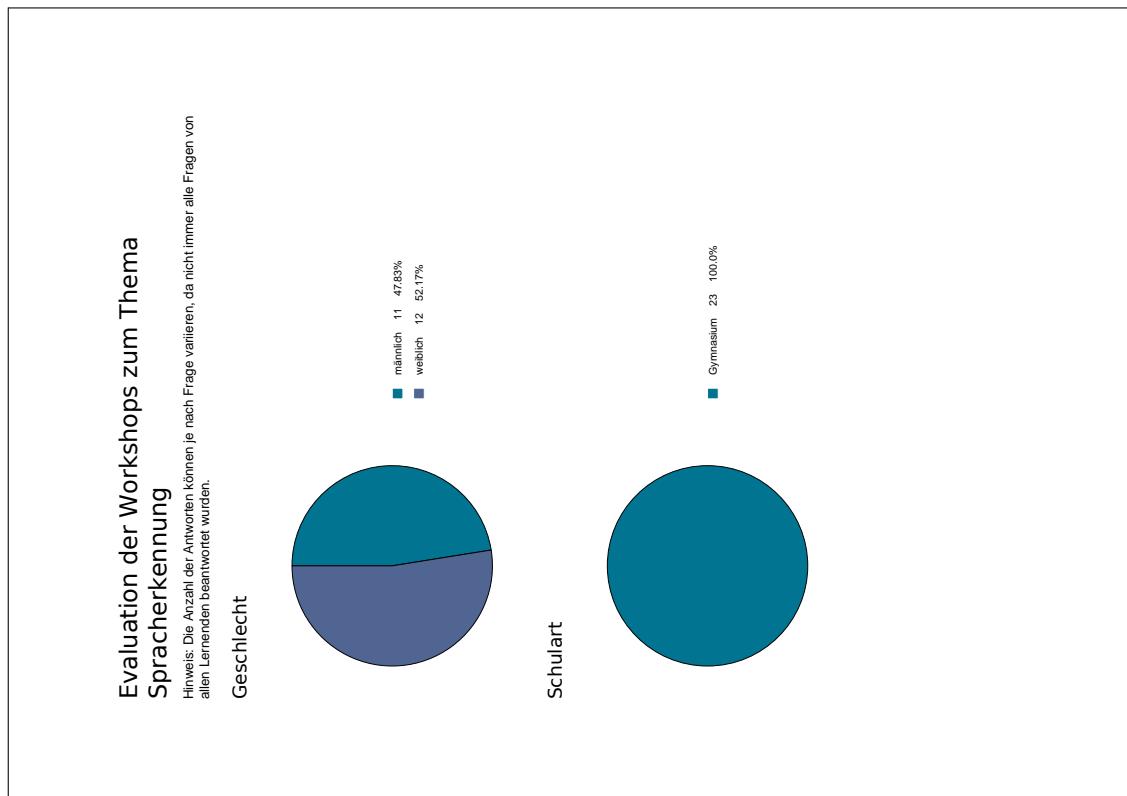
offene Aufgaben: Die Aufgaben haben mein Interesse am Thema geweckt.



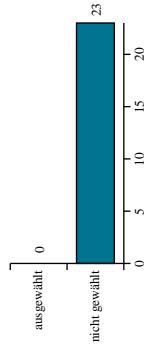
offene Aufgaben: Ich hätte mir mehr Aufgaben zum Programmieren gewünscht.



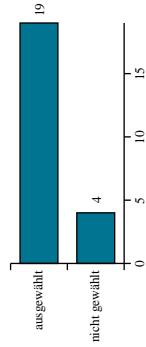
C.2 Evaluationen der Durchführungen zum Thema Spracherkennung



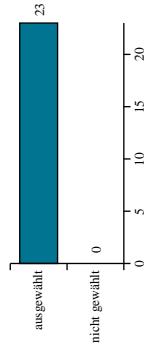
Leistungskurs: Informatik



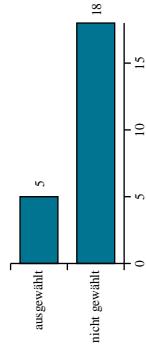
Leistungskurs: Andere



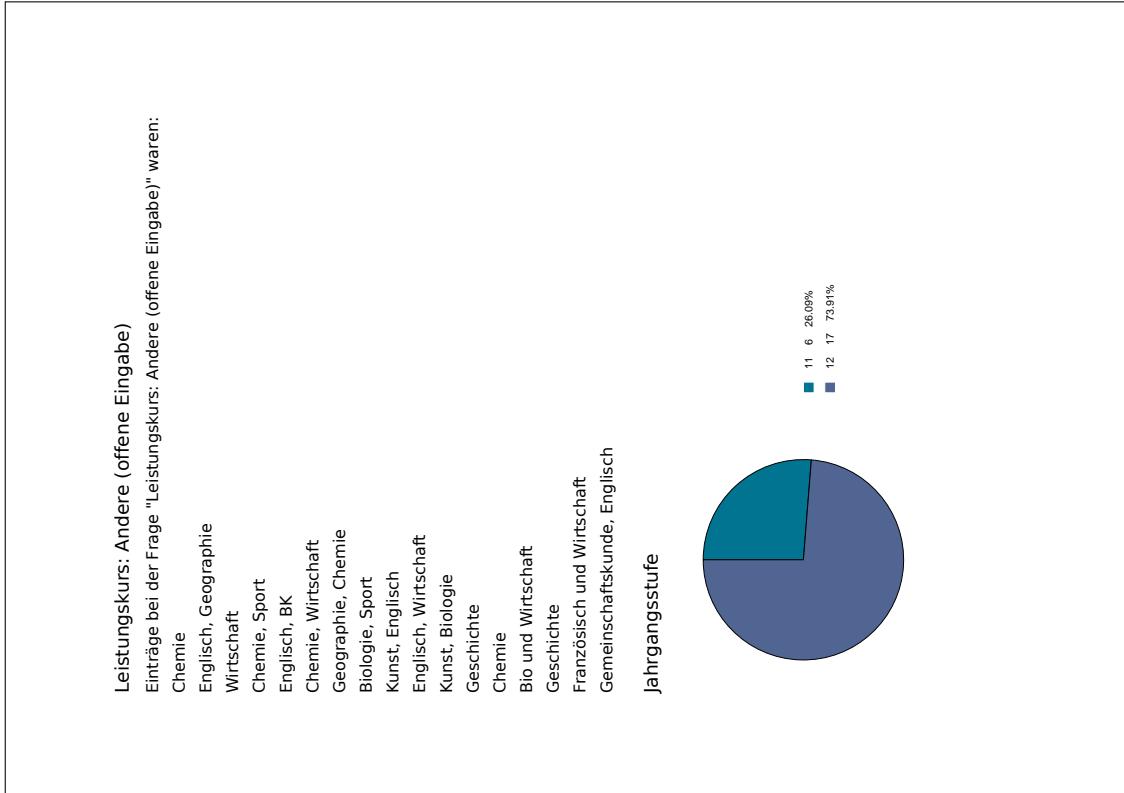
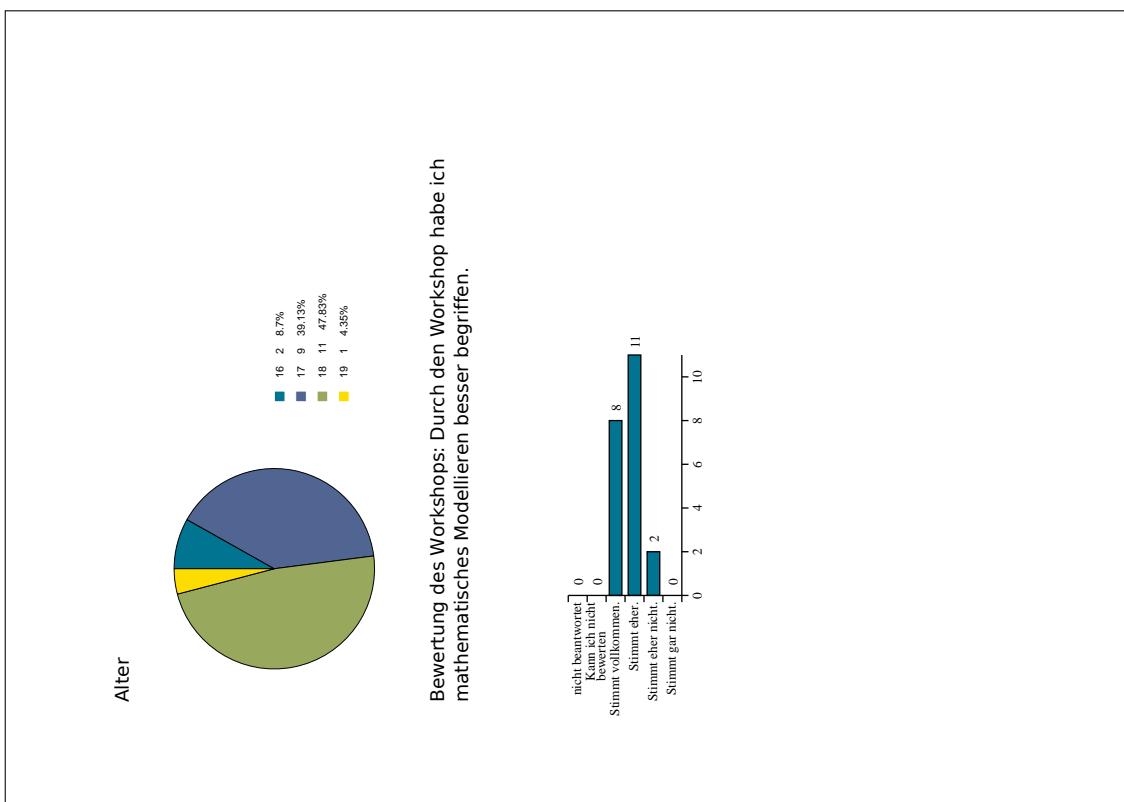
Leistungskurs: Mathematik



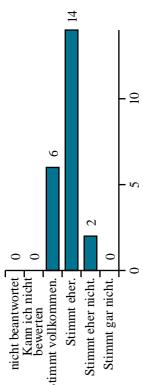
Leistungskurs: Physik



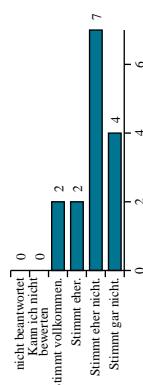
C.2 Evaluationen der Durchführungen zum Thema Spracherkennung



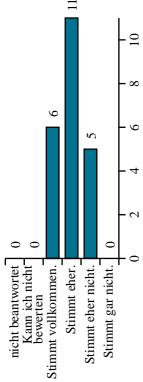
Bewertung des Workshops: Der Vortrag über mathematische Modellierung war hilfreich.



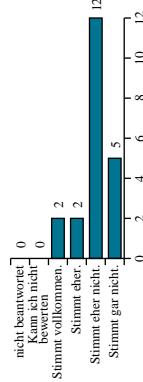
Bewertung des Workshops: Der Umgang mit der Programmiersprache Julia/Python fiel mir schwer.



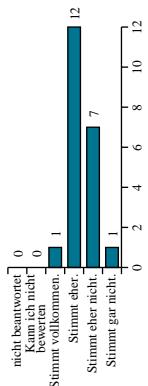
Bewertung des Workshops: Die Gestaltung der Arbeitsblätter fand ich gut.



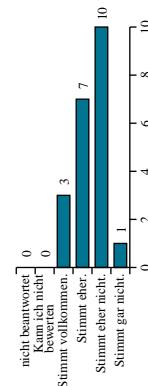
Bewertung des Workshops: Die Aufgaben waren zu einfach.



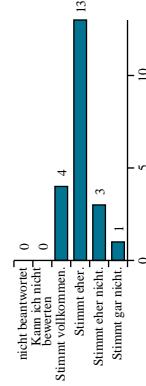
Bewertung des Workshops: Die Aufgaben waren zu schwierig.



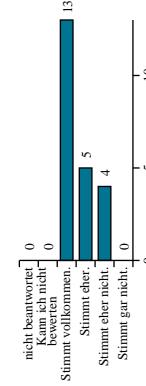
Bewertung des Workshops: Die Tipps / Hilfekarten waren hilfreich.



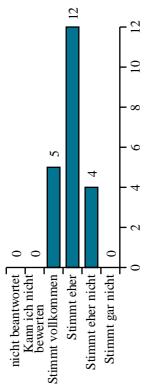
Bewertung des Workshops: Ich konnte die Aufgaben eigenständig bearbeiten.



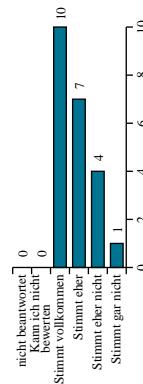
Bewertung des Workshops: Ich fand die automatischen Rückmeldungen, die ich nach der Eingabe meiner Lösungen im Code erhalten habe, hilfreich.



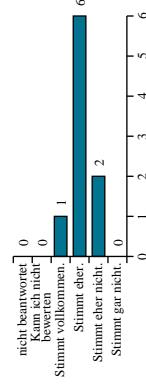
Weiterführende Fragen: Die Inhalte wurden klar vermittelt.



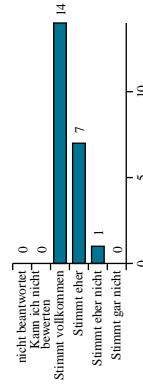
Weiterführende Fragen: Die Lern- und Arbeitszeiten waren angemessen (nicht zu lang oder zu kurz, mit ausreichenden Pausen).



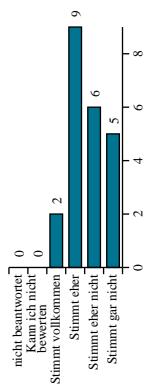
Bewertung des Workshops: Die Zusatzaufgaben haben mir sehr gefallen.



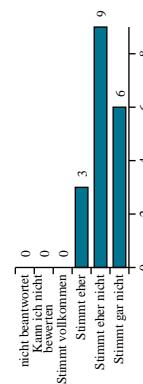
Weiterführende Fragen: Die Lern- und Arbeitsatmosphäre war angenehm.



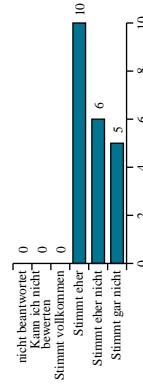
Weiterführende Fragen: Das Projekt hat mein Interesse an Themen der Naturwissenschaften und Technik gesteigert.



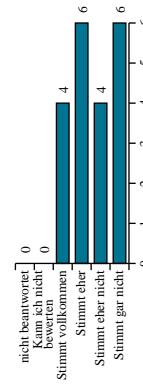
Weiterführende Fragen: Durch den Workshop habe ich interessante Berufs- und Studienmöglichkeiten kennengelernt.



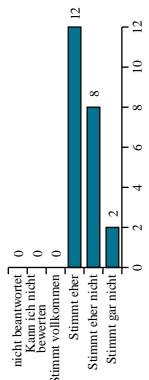
Weiterführende Fragen: Ich habe in diesem Workshop viel Neues gelernt, was mir für die Schule, für ein Studium und / oder Beruf weiterhelfen kann.



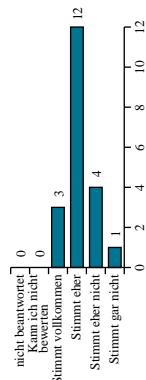
Weiterführende Fragen: Ich kann mir vorstellen ein Studium oder eine Ausbildung im Bereich der Naturwissenschaften oder der Technik zu beginnen.



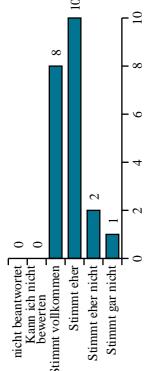
Weiterführende Fragen: Ich würde so einen Workshop gerne noch einmal besuchen.



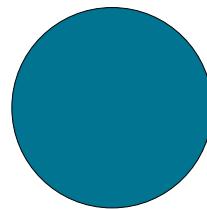
Weiterführende Fragen: Ich würde so einen Workshop auch anderen weiterempfehlen.

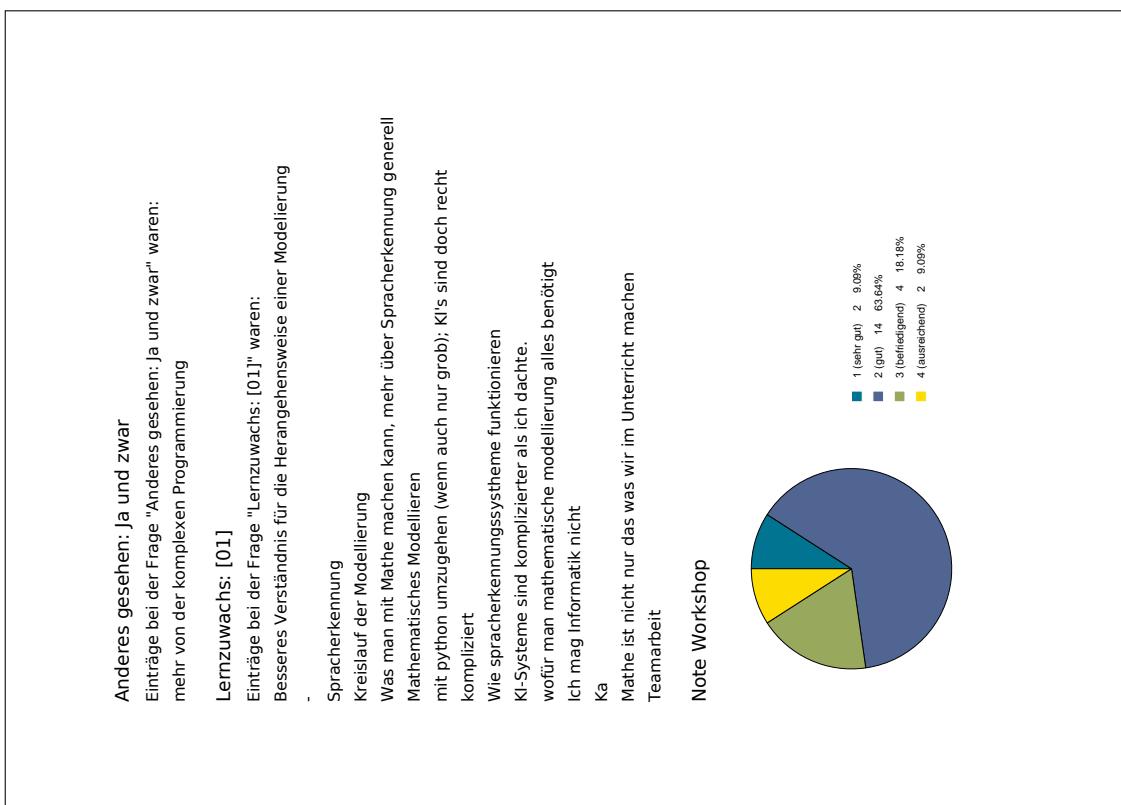


Weiterführende Fragen: Alles in allem hat mir der Workshop gut gefallen.

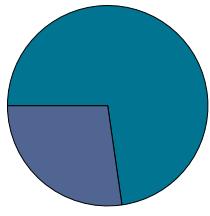


Nicht gefallen





Note Betreuung



Abschließender Kommentar: [01]

Einträge bei der Frage 'Abschließender Kommentar: [01]' waren:

Sehr freundliche und hilfreiche Betreuung

Ich fand den Vortrag und die Gestaltung gut. Ich hätte gedacht dass es noch mehr um KI bzw. alltagsbezogene Probleme geht.

Der Workshop war sehr interessant und es war eine entspannte

Atmosphäre

War gut

Sehr nette Aufsichtern, bei Fragen sofort zur Stelle und gute Darstellung durch Präsentationen; Vielleicht bessere Server, damit die Aufgaben bei jedem 'sefort' laden :)

Sehr netter Umgang

Noice{EM_THUMBS_UP}
{EM_OK_HAND}