

Multiwort-Bedeutungsauflösung bei Anforderungen

Bachelorarbeit
von

Thomas Bartel

An der Fakultät für Informatik
Institut für Programmstrukturen
und Datenorganisation (IPD)

Erstgutachter:	Prof. Dr. Walter F. Tichy
Zweitgutachter:	Prof. Dr. Ralf H. Reussner
Betreuender Mitarbeiter:	M.Sc. Tobias Hey

Bearbeitungszeit: 10.12.2019 – 09.04.2020

Kurzfassung

Zur automatischen Erzeugung von Rückverfolgbarkeitsinformationen muss zunächst die Absicht der Anforderungen verstanden werden. Die Grundvoraussetzung hierfür bildet das Verständnis der Bedeutungen der Worte innerhalb von Anforderungen. Obwohl hierfür bereits klassische Systeme zur Wortbedeutungsauflösung existieren, arbeiten diese meist nur auf Wortebene und ignorieren sogenannte „Multiwort-Ausdrücke“ (MWAs), deren Bedeutung sich von der Bedeutung der einzelnen Teilworte unterscheidet. Im Rahmen des INDIRECT-Projektes wird deshalb ein System entwickelt, welches die MWAs mithilfe eines einfach verketteten Zufallsfeldes erkennt und anschließend eine wissensbasierte Bedeutungsauflösung mit den Wissensbasen DBpedia und WordNet 3.1 durchführt. Um das System zu evaluieren wird ein Datensatz aus frei verfügbaren Anforderungen erstellt. Das Teilsystem für die Erkennung von MWAs erreicht dabei maximal einen F_1 -Wert von 0.81. Die Bedeutungsauflösung mit der Wissensbasis DBpedia erreicht maximal einen F_1 -Wert von 0.496. Mit der Wissensbasis WordNet 3.1 wird maximal ein F_1 -Wert von 0.547 erreicht.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	2
1.2	Aufbau der Bachelorarbeit	2
2	Grundlagen	3
2.1	Verarbeitung natürlicher Sprache	3
2.1.1	Tokenisierung	3
2.1.2	Lemmatisierung	4
2.1.3	Wortarten	4
2.1.4	Wortklassen	4
2.1.5	Wortartmarkierung	4
2.1.6	Das Penn Treebank POS Tagset	4
2.1.7	Syntaxbaum	5
2.1.8	Abhängigkeitsgraph	6
2.2	Wortbedeutungsrepräsentation	6
2.2.1	Synset	6
2.2.2	WordNet	7
2.2.3	Wikipedia und DBpedia	7
2.3	Maschinelles Lernen	8
2.4	Wortbedeutungsauflösung	8
2.4.1	Wissensbasierte Wortbedeutungsauflösung	9
2.4.2	Überwachte Wortbedeutungsauflösung	9
2.4.3	Unüberwachte Wortbedeutungsauflösung	10
3	INDIRECT	11
3.1	Vorgehen	11
3.2	INDIRECT im Rahmen dieser Arbeit	12
4	Verwandte Arbeiten	13
4.1	Erkennung und Extraktion von MWAs	13
4.2	Wortbedeutungsauflösung	15
5	Analyse und Entwurf	21
5.1	Ziele dieser Arbeit	21
5.2	Definition des Begriffs „Multiwort-Ausdruck“	22
5.2.1	Lexikalisierte Phrasen	22
5.3	MWAs im Kontext dieser Arbeit	23
5.4	Erkennung der MWAs	24
5.4.1	MWA-Erkennung anhand vorgefertigter Listen	25
5.4.2	MWA-Identifikation durch maschinelles Lernen	27
5.5	Wortbedeutungsauflösung	27
5.5.1	Wissensbasierte Wortbedeutungsauflösung	28

5.5.2	Überwachte Wortbedeutungsauflösung	30
5.5.3	Unüberwachte Wortbedeutungsauflösung	31
5.6	Wissensbasierte Bedeutungsauflösung bei MWAs	32
5.6.1	Wikipedia für die wissensbasierte Bedeutungsauflösung	33
5.6.2	Bedeutungsauflösung mit Berücksichtigung von MWAs	35
5.7	Wortbedeutungsauflösung durch UKB	36
5.8	Entwurf	37
5.8.1	Agent für die MWA - Erkennung	37
5.8.2	Agent für die Bedeutungsauflösung	39
6	Implementierung	41
6.1	Identifikation von MWAs basierend auf einem bedingten Zufallsfeld	41
6.1.1	Trainingsdatensatz	42
6.1.2	Agent für die MWA-Detektion	43
6.2	Wissensbasierte Bedeutungsauflösung	44
6.2.1	Agent für die Bedeutungsauflösung	47
7	Evaluation	51
7.1	Evaluationsdatensatz	51
7.1.1	Häufigste MWAs	54
7.2	Evaluationsmetriken	55
7.3	Evaluation der MWA-Erkennung	56
7.4	Evaluation der Bedeutungsauflösung	59
7.4.1	Evaluation ohne Teilbedeutungen	59
7.4.2	Evaluation mit Teilbedeutungen	63
7.4.3	Evaluation ohne Fehler der Wissensbasis	66
7.5	Kombinierte Evaluation	68
8	Zusammenfassung und Ausblick	71
	Literaturverzeichnis	73
	Anhang	79
A	Evaluationsergebnisse	79

Abbildungsverzeichnis

3.1	Aufbau des INDIRECT-Projektes	12
5.1	Agent für die MWA-Erkennung	38
5.2	Tokens nach der Verarbeitung durch den Agenten	38
5.3	Agent für die Bedeutungsauflösung	39
6.1	Klassendiagramm der Agenten für die MWA-Erkennung	43
6.2	MWA-Token mit Kindern	44
6.3	Klassendiagramm der Agenten für die Bedeutungsauflösung	47
7.1	Häufigste MWAs im Evaluationsdatensatz	54

Tabellenverzeichnis

2.1	Ausschnitt aus dem Penn Treebank POS Tagset [JM19]	5
5.1	Beispiele für semi-feste Ausdrücke [SBB ⁺ 02]	23
5.2	Beispiele für semi-feste Ausdrücke [SBB ⁺ 02]	23
6.1	Abbildung des Penn Treebank POS Tagsets auf die 17 Wortartmarkierungen [dim]	42
6.2	Auszug aus dem erstellten Trainingsdatensatz im Dimsum-Format	42
7.1	Auszug aus dem Evaluationsdatensatz	52
7.2	Statistiken über einzelne Wörter im Evaluationsdatensatz	53
7.3	MWA Statistiken des Evaluationsdatensatzes	54
7.4	Ergebnisse der 10-fachen Kreuzvalidierung ohne Teiltreffer	57
7.5	Ergebnisse der 10-fachen Kreuzvalidierung ohne Teiltreffer und mit Suche im Wörterbuch	58
7.6	Evaluation der DBpedia Graphen (ohne Teilbedeutungen/mit Fehlern der Wissensbasis)	60
7.7	Evaluation des WordNet 3.1 Graphen (ohne Teilbedeutungen/mit Fehlern der Wissensbasis)	62
7.8	Evaluation der DBpedia Graphen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)	63
7.9	Evaluation des WordNet 3.1 Graphen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)	65
7.10	Evaluation des WordNet 3.1 Graphen mit den ersten 100 Anforderungen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)	66
7.11	Evaluation der DBpedia Graphen (mit Teilbedeutungen/ohne Fehler der Wissensbasis)	66
7.12	Evaluation des WordNet 3.1 Graphen (mit Teilbedeutungen/ohne Fehler der Wissensbasis)	67
7.13	Vergleich der kombinierten Ergebnisse (DBpedia/mit Teilebedeutungen/mit Fehlern der Wissensbasis)	69
7.14	Vergleich der kombinierten Ergebnisse (WordNet/mit Teilbedeutungen/mit Fehlern der Wissensbasis)	70
A.1	Vergleich der kombinierten Ergebnisse (DBpedia/ohne Teilbedeutungen/- mit Fehlern der Wissensbasis)	79
A.2	Vergleich der kombinierten Ergebnisse (DBpedia/mit Teilbedeutungen/oh- ne Fehler der Wissensbasis)	79
A.3	Vergleich der kombinierten Ergebnisse (WordNet/ohne Teilbedeutungen/- mit Fehlern der Wissensbasis)	80
A.4	Vergleich der kombinierten Ergebnisse (WordNet/mit Teilbedeutungen/oh- ne Fehler der Wissensbasis)	80

1 Einleitung

Wartung und Pflege von Software ist eine Aufgabenstellung, die im Bereich der Softwaretechnik große Relevanz besitzt. Um diesen Prozess sowohl zu verbessern, als auch zu vereinfachen, können sogenannte Rückverfolgbarkeitsinformationen zwischen Anforderungen und Quelltext innerhalb eines Softwareprojektes genutzt werden [ACC⁺02]. Diese Informationen müssen derzeit jedoch noch mit hohem Aufwand manuell erstellt werden.

Damit dieser Prozess von einem System automatisiert werden kann, muss der Inhalt der Anforderungen verstanden werden. Eine Grundvoraussetzung hierfür ist es, die Bedeutung der einzelnen Wörter der Anforderungen zu kennen, denn nur durch diese kann erkannt werden, welche Entitäten innerhalb einer Anforderung agieren, oder wie die Entitäten einer Anforderung zueinander in Beziehung stehen. Erst dadurch entsteht ein tieferes Verständnis, welches dann für die Erzeugung von Rückverfolgbarkeitsinformationen genutzt werden kann. Diese Aufgabe übernehmen sogenannte Systeme zur Wortbedeutungsauflösung (engl. word sense disambiguation (WSD)), die Wörtern ihre im jeweiligen Kontext verwendete Bedeutung zuordnen. Viele dieser Systeme arbeiten allerdings nur auf einzelnen Wörtern. Das Problem hierbei jedoch ist, dass die in englischer Sprache verfassten Anforderungen häufig Fachbegriffe und zusammengesetzte Substantive (engl. compound nouns) enthalten, die aus mehreren Wörtern bestehen. Die Bedeutung solcher Ausdrücke, unterscheidet sich häufig von den Bedeutungen der einzelnen Teilworte, weshalb eine Bedeutungsauflösung auf Wortebene nicht sinnvoll ist. Neben weiteren Ausdrücken, die aus mehreren Wörtern bestehen, werden diese in dieser Arbeit durch den Begriff Multiwort-Ausdruck (MWA) zusammengefasst.

Beispiel 1.1: Beispiel für ein MWA

The network is unavailable because of a **denial of service attack**

Beispiel 1.1 zeigt ein zusammengesetztes Substantiv, dessen Bedeutung sich von der Bedeutung der einzelnen Teilworte unterscheidet. Mit dem Ausdruck „**denial of service attack**“ ist ein spezifischer Angriff zum Lahmlegen eines Netzwerks gemeint. Betrachtet man jedoch die Bedeutungen der einzelnen Teilworte wie „denial“, „service“ und „attack“, so ist nicht direkt ersichtlich, dass es sich um einen Angriff auf ein Netzwerk handelt. Die vollständige Bedeutung des MWAs kann also nur erkannt werden, wenn dieser als Ganzes betrachtet wird. Das Erkennen solcher MWAs und das korrekte Auflösen ihrer Bedeutung ist somit erstrebenswert.

Um Bedeutungen möglichst allgemeingültig abzubilden, nutzen WSD-Systeme Wissensquellen. Diese Wissensquellen sind jedoch unvollständig und hindern WSD-Systeme in einigen Fällen daran, einem Wort die richtige Bedeutung zuzuordnen. Da MWAs in vielen Fällen sehr spezifisch sind, besteht die Möglichkeit, dass eine verwendete Wissensquelle keinen Eintrag zu diesem enthält. Im Gegensatz zu einzelnen Wörtern, gehen bei einem MWA Informationen über mehrere Wörter verloren, falls für diesen kein Eintrag in der Wissensquelle vorhanden ist. Um dies zu verhindern kann versucht werden anhand der Bedeutungen der einzelnen Teilworte eines MWAs Rückschlüsse auf die Bedeutung des gesamten Ausdrucks zu ziehen. Verwendet man beispielsweise eine Wissensquelle, die keine Definition für den Ausdruck „**denial of service attack**“ enthält, so kann zumindest die Bedeutung des Wortes „**attack**“ genutzt werden, um zu erkennen, dass es sich um eine Art von Angriff handelt.

1.1 Zielsetzung

Das Ziel dieser Bachelorarbeit ist der Entwurf und die Evaluation eines WSD-Systems für das INDIRECT-Projekt [Hey19], dem es möglich ist MWAs zu erkennen und ihnen ihre korrekte Bedeutung in Abhängigkeit des gegebenen Kontexts zuzuordnen. Als Grundlage für die Bedeutungen sollen die Wissensbasen DBpedia [LIJ⁺15] und WordNet [Mil95] verwendet werden. Für den Fall, dass die korrekte Bedeutung in der genutzten Wissensquelle nicht vorhanden ist, soll ein Verfahren entwickelt werden, dass anhand der Bedeutungen der einzelnen Teilworte versucht Rückschlüsse auf die korrekte Bedeutung des MWAs zu ziehen. Der Fokus des entwickelten WSD-Systems soll auf Anforderungstexten liegen.

1.2 Aufbau der Bachelorarbeit

In Kapitel 2 werden zunächst, die zum Verständnis dieser Arbeit, notwendigen Grundlagen näher erläutert. Kapitel 3 beschreibt den Aufbau und die Funktionsweise des Projektes INDIRECT, in dessen Rahmen diese Arbeit verfasst wurde. In Kapitel 4 werden ausgewählte verwandte Arbeiten, die bereits Lösungen für Teilprobleme liefern, oder in anderer Art und Weise mit dieser Arbeit verwandt sind, erläutert. Die Ansätze der verwandten Arbeiten, verschiedene Lösungsansätze, Werkzeuge und Wissensbasen werden dann in Kapitel 5 näher analysiert und anhand dieser Analyseergebnisse wird anschließend eine Lösungsmethode für das Problem dieser Arbeit entworfen. Kapitel 6 beschreibt dann die genaue Implementierung der Lösungsmethode und alle verwendeten Werkzeuge. In Kapitel 7 wird eine Evaluation des implementierten Systems durchgeführt, die Ergebnisse analysiert und mögliche Verbesserungen für zukünftige Arbeiten genannt. Schlussendlich wird dann in Kapitel 8 eine kurze Zusammenfassung der Arbeit und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Grundlagen

Damit das Thema dieser Arbeit vollständig verstanden werden kann, ist es zunächst nötig, einige grundlegende Begriffe und Techniken zu erklären. In diesem Kapitel werden grundlegende Techniken der Verarbeitung von natürlicher Sprache, des maschinellen Lernens, sowie der Wortbedeutungsauflösung näher erklärt.

2.1 Verarbeitung natürlicher Sprache

Die natürliche Sprachverarbeitung (engl. natural language processing (NLP)) ist ein Teilgebiet der künstlichen Intelligenz und Linguistik, das sich mit dem Verständnis von menschlicher Sprache durch Computer beschäftigt. Die Hauptmotivation von NLP ist die Möglichkeit der Kommunikation zwischen Mensch und Maschine in natürlicher Sprache. Dies erlaubt es den Menschen das Erlernen einer Maschinensprache zu umgehen, da die Kommunikation mit Computern bereits in einer bekannten natürlichen Sprache möglich ist [KKKS17].

2.1.1 Tokenisierung

Bevor ein Text in natürlicher Sprache von einem System bearbeitet werden kann, müssen zunächst einige Normalisierungsprozesse durchgeführt werden. Einer dieser Prozesse nennt sich Tokenisierung (engl. tokenization). Hierbei werden laufende Texte in eine Menge von sogenannten Token segmentiert. Token stellen die kleinsten zu verarbeitenden Einheiten innerhalb eines Satzes dar und bestehen zumeist aus einzelnen Wörtern. Ein Datum im Format DD/MM/YY wird immer als ein gemeinsamer Ausdruck angesehen, weshalb es auch von einem einzigen Token zusammengefasst werden sollte. Des Weiteren ist die Tokenisierung kein einheitlicher Prozess und abhängig davon welche Sprache und welcher Standard verwendet werden. Es muss beispielsweise in Betracht gezogen werden, in welcher Art und Weise die Ziffern innerhalb einer Zahl getrennt werden. Manche Sprachen verwenden hierfür ein Komma, während andere einen Punkt benutzen [JM19].

Beispiel 2.1: Beispiel für einen tokenisierten Satz

- Satz: The user needs to be able to access the system properties.
- tokenisierter Satz: ['The', 'user', 'needs', 'to', 'be', 'able', 'to', 'access', 'the', 'system', 'properties', '.']

Beispiel 2.1 zeigt eine mögliche Tokenisierung eines Satzes. Die identifizierten Token bilden die Grundlage für einige Algorithmen bei der Verarbeitung von natürlicher Sprache.

2.1.2 Lemmatisierung

Die Lemmatisierung ist ein weiterer Normalisierungsprozess, der meistens vor der eigentlichen Textverarbeitung durchgeführt wird. Hierbei wird versucht die gemeinsame Stammform mehrerer unterschiedlicher Wörter zu finden. Diese Stammform wird als Lemma bezeichnet [JM19]. Dies ist deshalb relevant, da sich die Schreibweise bestimmter Wörter, abhängig vom Kontext, in dem sie verwendet werden, verändert. Ein Beispiel hierfür ist die Konjugation von Verben. In den Ausdrücken „Ich bin“, „Er ist“, „Sie sind“, besitzen die Verben alle die gemeinsame Stammform „sein“. Die Lemmatisierung erleichtert somit beispielsweise die Suche des Verbs „sein“ innerhalb eines Textes [JM19]. Die Lemmatisierung ist von der Normalformreduktion (engl. stemming), bei der lediglich alle morphologischen Inflektionen des Wortes entfernt werden, zu unterscheiden. Die beiden Prozesse sind jedoch in vielen Fällen ähnlich, da die Stammform eines Wortes häufig identisch ist zu dessen Normalform [WTLB18].

2.1.3 Wortarten

Als Wortarten (engl. part of speech (POS)) werden die syntaktischen Kategorien eines Wortes bezeichnet. Die syntaktische Kategorie eines Wortes zu kennen, ist ein wichtiger Bestandteil der Verarbeitung von natürlicher Sprache, da hiermit zusätzliche Informationen über das Wort und die Nachbarwörter gewonnen werden können. Beispielsweise kann bei einem Nomen häufig angenommen werden, dass es Teil einer Nominalphrase ist. Diese zusätzlichen Informationen helfen in vielen Teilbereichen der Verarbeitung von natürlicher Sprache wie zum Beispiel der Informationsextraktion, Spracherkennung, Sprachsynthese, etc. [JM19]

2.1.4 Wortklassen

Die verschiedenen Wortarten lassen sich generell darin unterscheiden, ob sie zu einer geschlossenen Klasse oder einer offenen Klasse gehören. Als geschlossene Klasse wird eine Menge von Wörtern bezeichnet, die im Allgemeinen nicht wächst und somit eine relativ feste Anzahl an Wörtern besitzt. Ein gutes Beispiel hierfür sind die Präpositionen, da eine Sprache nur sehr selten durch neue Präpositionen erweitert wird. Die Wörter die zu einer geschlossenen Klasse gehören treten generell häufig auf und werden zur grammatischen Strukturierung verwendet. Als Gegenstück dazu gibt es die offenen Klassen wie zum Beispiel die Verben, die ständig um neue Mitglieder erweitert werden. Als grobe Einteilung der offenen Klassen werden die Nomen, Verben, Adjektive und Adverbien festgelegt [JM19].

2.1.5 Wortartmarkierung

Als Wortartmarkierung (engl. part of speech tagging (POS-tagging)) wird der Prozess bezeichnet, den Bestandteilen eines Satzes ihre entsprechende Wortart zuzuordnen. Dies wird auf einem bereits tokenisierten Text durchgeführt, wobei jedes Token einer bestimmten Kategorie angehört. Das POS-tagging ist ein Disambiguierungsprozess, da Wörter oft mehreren möglichen Wortklassen angehören können [JM19]. Disambiguierungsprozesse sind generell nicht immer mit absoluter Genauigkeit möglich, weshalb die bisherigen Verfahren einem Wort nicht immer die korrekte Kategorie zuordnen können [McC09].

2.1.6 Das Penn Treebank POS Tagset

Das Penn Treebank POS Tagset ist eine Menge von Markierungen, die für die Bestimmung der verschiedenen Wortarten innerhalb eines Textkorpus häufig verwendet werden. Hierfür wird hinter jedem Wort ein Schrägstrich und daraufhin die entsprechende Markierung für die Wortart gesetzt.

Tabelle 2.1: Ausschnitt aus dem Penn Treebank POS Tagset [JM19]

Tag	Description	Example
DT	determiner	a, the
NN	sing or mass noun	llama
VBZ	verb 3sg pres	eats
TO	„to“	to
VB	verb base form	eat
JJ	adjective	yellow
NNS	noun, plural	llamas

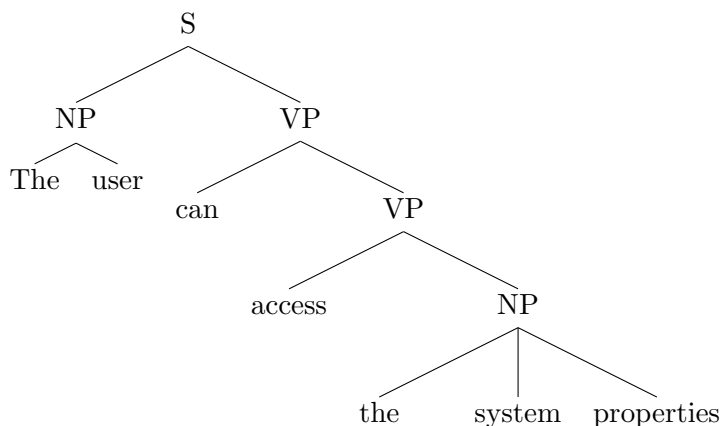
Beispiel 2.2: Beispielsatz mit Penn Treebank POS Tags

The/DT user/NN needs/VBZ to/TO be/VB able/JJ to/TO see/VB the/DT system/NN properties/NNS

Beispiel 2.2 zeigt einen Beispielsatz, bei dem die einzelnen Worte bereits eine Wortartmarkierung erhalten haben. Hierfür wurde ein Teil des Penn Treebank POS Tags, welcher in Tabelle 2.1 zu sehen ist, verwendet. Beispielsweise gibt der Ausdruck „needs/VBZ“ an, dass das Wort „needs“ ein Verb in der dritten Person Singular ist.

2.1.7 Syntaxbaum

Um die grammatische Struktur der englischen Sprache zu modellieren, ist ein formales System, wie eine Grammatik nötig. Ein möglicher Ansatz sind hierbei die Phrasenstrukturgrammatiken, welche die syntaktische Struktur von Phrasen anhand von Regeln festlegen. Am häufigsten wird hierfür eine kontextfreie Grammatik, mit einer Menge von Terminalsymbolen, Nichtterminalsymbolen und Produktionsregeln, verwendet. Aus jedem abstrakten Nichtterminalsymbol kann anhand der Produktionsregeln eine Menge von Terminal- und Nichtterminalsymbolen abgeleitet werden. Eine solche Grammatik kann dabei sowohl für die Strukturierung eines bereits vorhandenen Satzes, als auch für das Erzeugen neuer Sätze verwendet werden. Die Ableitungen aus den Nichtterminalsymbolen können anhand eines sogenannten Syntaxbaums dargestellt werden. Die Form dieses Syntaxbaumes ist abhängig von der gewählten Grammatik, mit der ein Satz analysiert wird [JM19].

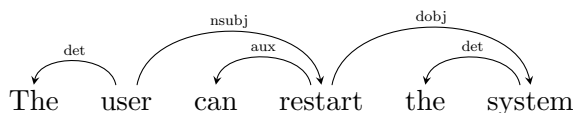
Beispiel 2.3: Syntaxbaum eines Satzes

In Beispiel 2.3 ist ein Syntaxbaum mit den Nichtterminalsymbolen S, NP und VP zu sehen. S ist das Startsymbol für einen Satz, NP steht für eine Nominalphrase und VP für eine Verbalphrase. Aus den Nichtterminalsymbolen NP und VP werden anschließend die einzelnen Worte des Satzes, welche alle Terminalsymbole darstellen, abgeleitet.

2.1.8 Abhängigkeitsgraph

Bei den bereits zuvor erwähnten kontextfreien Grammatiken existieren bestimmte Regeln, die den Aufbau von Phrasen genauer beschreiben. Diese Regeln nehmen eine wichtige Rolle bei der Modellierung einer Sprache ein. Es existieren jedoch auch Familien von Grammatiken, die weniger Wert auf die genaue Struktur von Phrasen legen. Ein Beispiel hierfür sind die Abhängigkeitsgrammatiken, die die syntaktische Struktur eines Satzes ausschließlich anhand von paarweisen Relationen zwischen Wörtern oder Lemmata innerhalb eines Satzes beschreiben. Der Vorteil der Abhängigkeitsgrammatiken zeigt sich vor allem bei Sprachen, deren Wortreihenfolge nicht durch strenge Regeln festgelegt ist. Diese Ambiguität lässt sich nur schwer durch eine Phrasenstrukturgrammatik modellieren, da jeder mögliche Fall durch eine Regel festgelegt werden muss. Abhängigkeitsgrammatiken haben dieses Problem nicht, da die Relationen zwischen den Wörtern unabhängig von ihrer Position innerhalb des Satzes modelliert werden können. Die Relationen der Wörter werden häufig anhand eines Abhängigkeitsgraphen dargestellt [JM19].

Beispiel 2.4: Beispiel für einen Abhängigkeitsgraphen



Beispiel 2.4 zeigt ein Beispiel für einen Abhängigkeitsgraphen, der aus den paarweisen Relationen der Wörter eines Satzes erstellt wurde. Daraus kann beispielsweise abgelesen werden, dass das Wort „can“ das zugehörige Hilfsverb des Wortes „restart“ ist.

2.2 Wortbedeutungsrepräsentation

Die meisten Wörter einer Sprache besitzen viele mögliche Bedeutungen, die vom Kontext, in dem das Wort auftritt, abhängig sind. Damit Systeme effizient, mit den Bedeutungen von Worten arbeiten können, muss zunächst eine effektive Repräsentation für diese gefunden werden. In diesem Zusammenhang gibt es einige Begriffe, die im Bereich der Wortbedeutungsrepräsentation immer wieder auftauchen.

2.2.1 Synset

Ein Synset stellt eine Menge von Synonymen dar. Alle Wörter innerhalb eines Synsets besitzen somit dieselbe Wortbedeutung. Jedes Synset wird in einer Wissensquelle auf genau eine Bedeutung abgebildet, weshalb die Identifikation des entsprechenden Synset eines Wortes gleichbedeutend ist mit der Wortbedeutungsauflösung [WTLB18].

Beispiel 2.5: Synsets aus der Wissensquelle WordNet [Mil95]

- Leaf, leafage, folio - the main organ of photosynthesis in higher plants
- Leaf, folio - a sheet of written or printed matte

In Beispiel 2.5 sind zwei Beispiele für die Wörter innerhalb eines Synsets und deren geteilte Bedeutung aufgelistet. Beispielsweise gehören die Wörter „leaf“, „leafage“ und „folio“ alle zum gleichen Synset und teilen sich somit die entsprechende Bedeutung: „the main organ of photosynthesis in higher plants“ [Mil95].

2.2.2 WordNet

WordNet ist eine öffentliche Wissensquelle, die in ihrer Funktion einem Wörterbuch gleicht und aufgrund der großen Menge an semantischen Informationen, die sie zur Verfügung stellt, häufig im Bereich der Wortbedeutungsauflösung verwendet wird [McC09]. In WordNet werden Einträge als Paare (f, s) von Form und Bedeutung gespeichert. Die Form f ist dabei eine Buchstabenfolge aus einem Alphabet und die Bedeutung s ein Element aus einer Menge von Bedeutungen. Eine Buchstabenfolge wird als Wort einer Sprache angesehen, falls ihr eine entsprechende Bedeutung in der Sprache zugewiesen werden kann [Mil95].

Für jedes Wort in WordNet wird beachtet, dass es in Abhängigkeit vom Kontext zu verschiedenen syntaktischen Kategorien zugeordnet werden kann. Sie werden deshalb eingeteilt in die Kategorien Nomen, Verb, Adjektiv und Adverb. Des Weiteren treten bei vielen Worten morphologische Inflektionen auf, wie beispielsweise bei der Konjugation von Verben. Diese verändern zwar die Form des Wortes, jedoch nicht die eigentliche Bedeutung. WordNet beachtet dies und gibt die entsprechenden Informationen über die Stammform des Wortes zurück [McC09].

Zwischen Wörtern können außerdem sogenannte semantische Relationen definiert werden. WordNet benutzt hier als Standardrelation die Synonymie und stellt Wortbedeutungen in Form von Synsets dar. Weitere verwendete Relationen sind beispielsweise die Antonymie, also Wörter mit gegensätzlicher Bedeutung, so wie Hyponymie und Hypernomie, also Unterbegriffe und Überbegriffe. Diese werden alle in Form von Zeigern zwischen verschiedenen Synsets dargestellt [Mil95].

2.2.3 Wikipedia und DBpedia

Wikipedia ist eine öffentliche Wissensquelle, die kollaborativ von vielen Menschen jeden Tag erweitert wird. Obwohl Wikipedia eigentlich eine textbasierte Ressource ist, werden dennoch seit geraumer Zeit viele strukturierte Daten, wie zum Beispiel Koordinaten und Taxonomien, gesammelt. Diese könnten potenziell wichtige Beiträge zu vielen Bereichen der Wissenschaft liefern. Ein direkter Zugriff auf die Daten ist jedoch in vielen Fällen nicht direkt durch Wikipedia möglich, was die Nutzung deutlich erschwert. Um dieses Problem zu lösen wurde das Projekt DBpedia gestartet, welches die strukturierten Daten von Wikipedia in 111 Sprachen extrahiert [LIJ⁺15].

Obwohl Wikipedia hauptsächlich aus Freitext besteht, sind dennoch einige strukturierte Informationen wie Infoboxen, Links, Koordinaten, etc. vorhanden. Diese Daten werden mithilfe eines Extraktionsframeworks extrahiert und anschließend in Form von RDF Aussagen umgewandelt [LIJ⁺15].

Eine RDF Aussage wird verwendet, um Beziehungen zwischen Entitäten darzustellen und wird auch Subjekt-Prädikat-Objekt Ausdruck genannt [JM19].

Beispiel 2.6: Beispiel für ein RDF triple [JM19]

Subjekt	Prädikat	Objekt
Golden Gate Park	location	San Francisco

Beispiel 2.6 zeigt ein RDF triple, dass für die Darstellung einer Entitätenrelation verwendet wird. Das Subjekt „Golden Gate Park“ wird anhand des Prädikats „location“, dem Objekt „San Francisco“ zugewiesen. Somit erkennt man, dass der Golden Gate Park in San Francisco liegt.

Für jeden Artikel in DBpedia existieren einheitliche Ressourcenbezeichner (engl. uniform resource identifier (URI)), die für die Repräsentation von Konzepten verwendet werden. Jedes Konzept mit dem Präfix „http://dbpedia.org/resource/“ repräsentiert einen Artikel. Konzepte mit dem Präfix „http://dbpedia.org/property/“ repräsentieren Informationen, die aus Infoboxen extrahiert wurden. Der Präfix „http://dbpedia.org/ontology/“ wird verwendet, um die Ontologie von DBpedia zu repräsentieren [LIJ⁺15].

Die erzeugten RDF Aussagen können anschließend verwendet werden, um gezielt auf die strukturierten Informationen von Wikipedia und den zugehörigen Relationen zugreifen zu können.

2.3 Maschinelles Lernen

Das maschinelle Lernen ist ein Forschungsbereich, der sich damit beschäftigt, wie ein Computer das Erfüllen bestimmter Aufgaben erlernen kann. Hierzu zählen beispielsweise Aufgaben wie das Erkennen von Zeichen oder die Trennung von Materialien anhand bestimmter Charakteristiken. Ein System, welches lernen kann, wie die genannten Aufgaben zu erfüllen sind, hat die Möglichkeit Experten in verschiedenen Arbeitsbereichen bei ihren täglichen Aufgaben zu unterstützen. Generell unterscheidet man beim maschinellen Lernen zwischen den überwachten und den unüberwachten Lernverfahren [MAPS18].

Die überwachten Lernverfahren haben als Ziel, einem System beizubringen, wie es einem Eingabedatum die korrekte Klasse aus einer Menge von Klassen zuordnet. Hierfür muss zunächst ein Trainingsdatensatz, dessen Daten bereits manuell mit einer Klasse markiert wurden, erstellt werden [MAPS18].

Bei den unüberwachten Lernverfahren soll das System hingegen lernen Eingabedaten zu analysieren, um so bestimmte relevante Gruppierungen zu finden [MAPS18].

2.4 Wortbedeutungsauflösung

Die Wortbedeutungsauflösung ist ein Teilgebiet der Verarbeitung von natürlicher Sprache (engl. natural language processing (NLP)), bei dem Systeme versuchen, die korrekte Bedeutung eines Wortes, anhand des Kontextes in dem es auftritt, zu bestimmen [McC09]. WSD ist damit unerlässlich für Systeme, deren Funktionalität von dem Verständnis von natürlicher Sprache abhängt. Ein Beispiel hierfür ist die maschinelle Übersetzung von Sprachen. Wörter besitzen häufig viele verschiedene mögliche Übersetzungen innerhalb einer Sprache. Damit eine korrekte Übersetzung in eine andere Sprache möglich ist, muss zunächst die Bedeutung des Wortes innerhalb eines Satzes festgestellt werden [IV98]. Die Aufgaben der Wortbedeutungsauflösung werden in zwei verschiedene Bereiche eingeteilt:

Lexical sample WSD task

Bei dieser Aufgabe ist eine Menge an ausgewählten Zielwörtern vorgegeben, die bei der Analyse eines Eingabetextes disambiguiert werden müssen [JM19]. Allen anderen Wörtern muss keine Bedeutung zugeordnet werden.

All-words WSD task

Diese Aufgabe ist schwieriger als die lexical sample Aufgabe, da hierbei keine Menge von Zielwörtern vorgegeben wird. Für jedes Wort des Eingabetextes muss eine Wortbedeutungsauflösung durchgeführt werden [JM19].

Grundsätzlich lässt sich die Lösung dieser beiden Aufgaben der Wortbedeutungsauflösung anhand von drei verschiedenen möglichen Ansätzen kategorisieren. Diese besitzen alle verschiedene Vor- und Nachteile und es ist situationsbedingt, welcher Ansatz die richtige Wahl ist.

2.4.1 Wissensbasierte Wortbedeutungsauflösung

Die wissensbasierte Bedeutungsauflösung versucht Informationen aus vorhandenen Wissensquellen zu extrahieren, um diese zur Wortbedeutungsauflösung zu nutzen. Diese Wissensquellen treten in verschiedenen Formen, wie z.B. Wörterbücher, Lexika, etc. auf, und müssen alle unter erheblichem manuellen Aufwand erstellt werden. Die Güte dieses Verfahrens ist abhängig davon, ob die Wissensquelle eine große Breite an Bedeutungen für die Wörter bereitstellt. Das Verfahren ist deshalb zuverlässig für die meisten allgemeinen Kontexte eines Wortes. Innerhalb von domänenspezifischen Texten besteht jedoch die Möglichkeit, dass Wörter auftreten, deren Definitionen nicht in Wissensquellen vorhanden sind. In diesem Fall ist eine Bedeutungsauflösung nicht möglich [McC09].

Beispiel 2.7: Definitionen des Wortes „compatibility“ in WordNet [WorB]

1. S: (n) compatibility (a feeling of sympathetic understanding)
2. S: (n) compatibility (capability of existing or performing in harmonious or congenial combination)

Um das Vorgehen zu verdeutlichen, sind in Beispiel 2.7 die Definitionen des Wortes „compatibility“, die in der bereits beschriebenen Wissensquelle WordNet vorhanden sind, aufgeführt. Bei einer wissensbasierten Bedeutungsauflösung wird dann der Kontext des Wortes betrachtet, um eine passende Definition aus der Liste von Definitionen auszuwählen. Lesk [Les86] schlägt dafür vor, die Definitionen des betrachteten Wortes und die Definitionen des Kontextes, in dem das betrachtete Wort auftritt, zu vergleichen. Dabei wird die Anzahl der Überlappungen zwischen den Definitionen gezählt. Die Definitionen, mit den meisten Überlappungen werden dann als wahrscheinlich angesehen. Es existieren jedoch auch einfache heuristische Ansätze. Dabei kann man einem Wort beispielsweise immer die am häufigsten auftretende Bedeutung zuweisen. [RPS15].

2.4.2 Überwachte Wortbedeutungsauflösung

Bei der überwachten Wortbedeutungsauflösung werden maschinelles Lernen und manuell angefertigte, mit Bedeutungen annotierte Textkorpora verwendet, um dem System die entsprechende Bedeutung eines Wortes in Abhängigkeit von bestimmten Kontexten beizubringen. Hierfür werden zunächst alle relevanten Wörter innerhalb eines Textkorpora manuell mit einer Bedeutung annotiert. Diese Annotationen werden alle auf eine Definition des Wortes in einer vorhandenen Wissensquelle abgebildet. Das lernende WSD-System hat so die Möglichkeit sofort die korrekte Bedeutung eines Wortes in dem gegebenen Textkorpora zu identifizieren. Ein oder mehrere dieser Textkorpora werden deshalb als Trainingsdatensatz für einen Klassifizierer verwendet [RPS15].

Beispiel 2.8: Ein Satz in dem das Wort „Plants“ mit einer Bedeutung annotiert ist [McC09]

```
<tag>Plants=_flora_</tag> and animals in this climate exhibit various defenses for the harsh conditions.
```

In Beispiel 2.8 wird der Nutzen der Bedeutungsannotation von Worten deutlich. Durch den vergebenen Tag, kann das System eindeutig erkennen, dass mit dem Wort „Plants“ in dem Satz eine Pflanze gemeint ist. Dies bietet dem System die Möglichkeit zu erlernen, dass zum Beispiel das Auftreten von „animals“ ein guter Hinweis für die pflanzliche Bedeutung des Wortes „Plants“ ist [McC09].

2.4.3 Unüberwachte Wortbedeutungsauflösung

Die unüberwachte Wortbedeutungsauflösung unterscheidet sich deutlich von den wissensbasierten und überwachten Ansätzen. Diese versuchen Informationen aus manuell angelegten Wissensquellen und mit Bedeutungen annotierten Textkorpora zu ziehen, um eine Wortbedeutungsauflösung durchzuführen. Dadurch sind sie beschränkt auf die Informationen, die bereits von Menschen erarbeitet wurden. Unüberwachte WSD-Systeme arbeiten jedoch mit Textkorpora, die nicht von Menschen manuell mit Bedeutungen annotiert wurden. Da keine Wortannotationen vorgegeben sind, muss das System zunächst die Informationen des Textkorpus nutzen, um die verschiedenen Wortbedeutungen zu identifizieren [RPS15]. Dieser Vorgang nennt sich die Wortbedeutungsinduktion (engl. word sense induction (WSI)) [McC09]. Anschließend können die induzierten Bedeutungen zur Wortbedeutungsauflösung verwendet werden.

Ein möglicher Ansatz diese Bedeutungen zu induzieren ist es, Wörter anhand ihrer vorhandenen Kontexte in Clustern zusammenzufassen. Diese Cluster werden anschließend als Repräsentanten für Wortbedeutungen verwendet.

Beispiel 2.9: Beispielcluster für das Wort „heart“ nach Pantel und Lin [PL02]

heart

- Nq72 0.27 (kidney, bone marrow, marrow, liver)
- Nq866 0.17 (psyche, consciousness, soul, mind)

Pantel und Lin [PL02] erstellen Cluster von Wörtern anhand des Kontextes, in dem sie auftreten. Beispiel 2.9 zeigt zwei Beispielcluster für das Wort „heart“. Das erste Cluster stellt dabei die Bedeutung des Organs dar, während das zweite Cluster eher eine psychologische Bedeutung repräsentiert.

3 INDIRECT

Diese Arbeit ist Teil des Projektes „INDIRECT: Intent-driven Requirements-to-Code Traceability“ [Hey19]. Rückverfolgbarkeitsinformationen zwischen den Anforderungen in natürlicher Sprache und dem Quelltext, sind relevant für die Wartung, Analyse und für diverse weitere Aufgaben bei Softwareprojekten. Obwohl diese Informationen in den genannten Bereichen eine wichtige Rolle spielen, werden sie vor allem wegen des hohen manuellen Aufwands nur selten generiert. INDIRECT bietet deshalb einen Ansatz die Anforderungen in natürlicher Sprache mit den entsprechenden Teilen des Quelltextes zu verknüpfen und somit Rückverfolgbarkeitsinformationen automatisch zu generieren. Um dies zu ermöglichen werden sowohl die Anforderungen, als auch der Quelltext in Form von Graphen modelliert, um so die Absichten beider Bestandteile zu repräsentieren.

3.1 Vorgehen

Abbildung 3.1 zeigt den groben Aufbau des INDIRECT-Projektes. Das Projekt beinhaltet sowohl ein Anforderungsmodell, als auch ein Quelltextmodell, die die Intentionen beider Informationsquellen modellieren sollen. Für das Anforderungsmodell soll mithilfe eines iterativen Ansatzes, bei dem mehrere Analyseschritte zum Verständnis von natürlicher Sprache (engl. natural language understanding (NLU)) verwendet werden, ein Graph erstellt werden, der die semantischen Beziehungen zwischen den Elementen der Anforderungen in natürlicher Sprache repräsentiert. Beispiele hierfür sind die syntaktische Analyse, um Verbindungen zwischen Aussagen und Entitäten zu schaffen, Wissenserweiterung, um den erstellten Graph mit Wissen aus bereits vorhandenen Wissensbasen zu füllen, oder auch die Konzeptualisierung, um das Modell mit Informationen bezüglich zugrunde liegender Konzepte zu erweitern.

Das Quelltextmodell wird, ähnlich wie das Anforderungsmodell, in Form eines Graphen modelliert, jedoch repräsentiert dieser die Beziehungen zwischen Bestandteilen des Quelltextes. Für den Aufbau des Modells werden sowohl strukturelle Informationen aus dem Quelltext selbst, als auch semantische Informationen aus zusätzlichen Quellen wie Kommentare, Dokumentationen und Commits verwendet. Für die strukturellen Informationen wird vor allem der Quelltext selbst untersucht, um herauszufinden welche Teile des Quelltextes eine semantische Verwandheit aufweisen. Diese Informationen werden mit den semantischen Informationen aus der Dokumentation und den Kommentaren kombiniert.

Beide Modelle werden anschließend verwendet um automatische Rückverfolgbarkeitsinformationen zwischen Anforderungen und Quelltext zu erstellen. Hierfür werden Muster

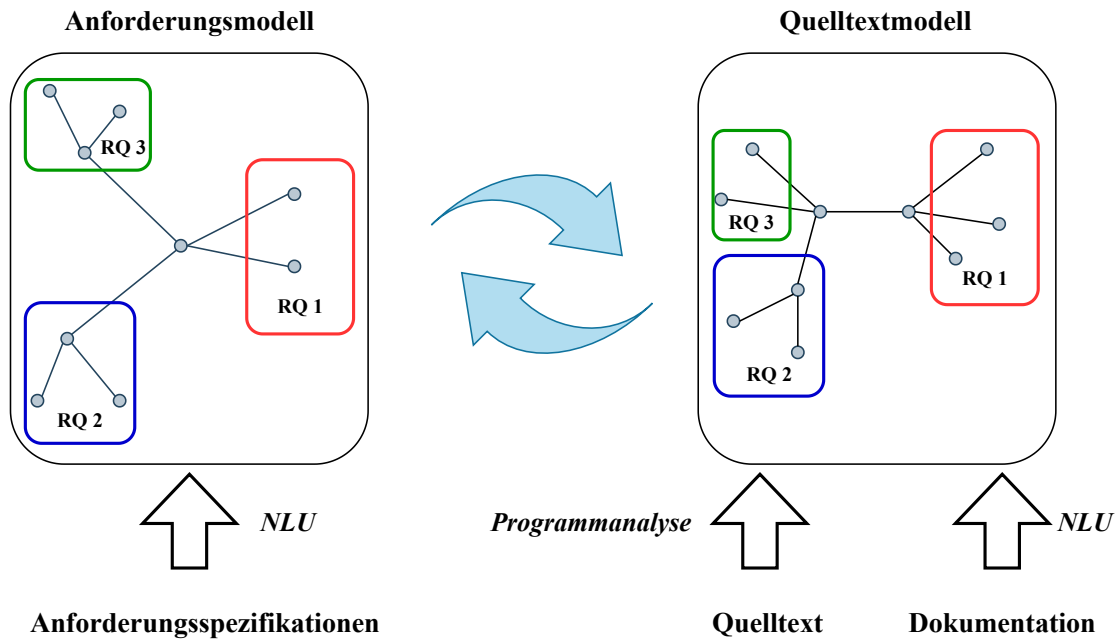


Abbildung 3.1: Aufbau des INDIRECT-Projektes

erlernt, die Ähnlichkeiten zwischen den Absichten, die von beiden Modellen repräsentiert werden, aufzeigen.

3.2 INDIRECT im Rahmen dieser Arbeit

Das zu entwickelnde WSD-System wird vor allem auf der Anforderungsseite des INDIRECT-Projektes angewendet. Die Informationen, die durch die Analyse der Anforderungsspezifikationen bereits im Anforderungsmodell vorhanden sind, werden teilweise von dem WSD-System verwendet. Damit sind vor allem die Informationen gemeint, die durch die syntaktische Analyse entstehen, wie beispielsweise die Tokenisierung, Wortartmarkierungen und Lemmata.

Wichtig ist außerdem, dass INDIRECT ein Framework aus Agenten bereitstellt, die verschiedene Aufgaben des Projektes übernehmen. Jeder Agent bekommt dabei Zugriff auf einen zentralen Graphen, dessen Informationen von den jeweiligen Agenten genutzt werden, um eine spezifische Aufgabe durchzuführen. Die Implementierung des WSD-Systems, soll ebenfalls innerhalb des Frameworks von INDIRECT geschehen.

4 Verwandte Arbeiten

In dem folgenden Kapitel werden verwandte Arbeiten näher beschrieben. Diese lassen sich grundsätzlich in zwei Teilgebiete der Verarbeitung natürlicher Sprache aufteilen. Das erste Teilgebiet beschäftigt sich mit der Erkennung und Extraktion von MWAs in Texten. Das zweite Teilgebiet beinhaltet die allgemeinen Ansätze der Wortbedeutungsauflösung bei Texten.

4.1 Erkennung und Extraktion von MWAs

Im nachfolgenden Abschnitt werden Arbeiten vorgestellt, die sich mit der Erkennung und Extraktion von MWAs beschäftigen. Dies ist ein wichtiger Bestandteil dieser Arbeit, da eine Wortbedeutungsauflösung bei MWAs erst möglich ist, wenn diese erkannt werden. Die beschriebenen Systeme wurden alle bei der zehnten Aufgabe der SemEval-2016 [SHJC16] evaluiert. Dabei sollen die eingereichten Systeme in einem Testkorpus minimale semantische Einheiten identifizieren und ihnen anschließend eine semantische Kategorie aus einer festen Menge zuordnen. Das Testkorpus enthält dabei eine Vielzahl an MWAs, die als zusammengehörige semantische Einheit markiert werden müssen [SHJC16].

In der Arbeit „UW-CSE at SemEval-2016 Task 10: Detecting Multiword Expressions and Supersenses using Double-Chained Conditional Random Fields“ [HSL16] stellen die Autoren Hosseini, Smith und Lee ihr System für die Identifikation von MWAs und der gleichzeitigen Markierung der semantischen Kategorie vor, welches sie bei der SemEval-2016 eingereicht haben. Das System hat die Aufgabe jedem Wort eines Korpus die Kategorien: „einzelnes Wort“, „erstes Wort eines MWA“, oder „Wort das ein MWA fortführt“ zuzuordnen. Für jedes Nomen und Verb gibt es außerdem eine Menge von semantischen Kategorien, die ihnen zugeordnet werden sollen. Um sowohl die Bezeichner für die semantische Kategorie, als auch die MWA-Bezeichner in einem einheitlichen Bezeichner zu kodieren, werden alle möglichen Paare der beiden Mengen gebildet. Die Autoren schlagen für ihr System die Nutzung eines doppelt verketteten bedingten Zufallsfeldes (engl. double-chained conditional random field) vor, das die bedingte Wahrscheinlichkeit für das Auftreten der Bezeichnerkombination $(\mathbf{m}, \mathbf{s})$ bei einem gegebenen Satz $\mathbf{x}$ voraussagt. Das $\mathbf{m}$ steht hierbei für eine Sequenz von MWA-Bezeichnern und das $\mathbf{s}$ für eine Sequenz von Bezeichnern für die semantischen Kategorien. Dies erlaubt die separate Modellierung der Merkmale der MWA-Bezeichner und der Bezeichner für die semantischen Kategorien. Das System wurde für die Evaluation bei der SemEval-2016 in zwei verschiedenen Teilkategorien eingereicht.

Da beide Kategorien unterschiedliche Einschränkungen bezüglich der erlaubten Ressourcen vorgeben, wurden für beide Kategorien unterschiedliche Eingabemerkmale verwendet. In der geschlossenen Kategorie durfte der bereits bereitgestellte Datensatz, das WordNet-Lexikon, ein bereitgestelltes Brown clustering des „Yelp Academic Dataset“ und die „ARK Tweet NLP cluster“ verwendet werden. Die offene Kategorie erlaubte jede verfügbare Ressource. Als Eingabemerkmale der offenen Kategorie wurden deshalb Lemmas, POS Tags, Wortformen, die bereitgestellten Brown Cluster, Merkmale bezüglich der Großschreibung, ein Merkmal für die Unterscheidung zwischen Hilfsverb und Hauptverb und Merkmale zu den semantischen Kategorien von WordNet verwendet. Außerdem wurde ein Merkmal verwendet, dass die Zugehörigkeit eines Tokens zu einem Eintrag in mehreren MWA-Lexika kennzeichnet. Diese Eingabemerkmale wurden anschließend mit den MWAs oder den semantischen Kategorien gekoppelt. Bei der geschlossenen Kategorie wurden alle Merkmale, die auf MWA-Lexika basieren, ausgelassen, da deren Nutzung in dieser Kategorie untersagt ist. Den besten Wert erzielte das System bei der Evaluation unter offenen Bedingungen mit einem kombinierten F_1 -Wert von 57.71% [SHJC16]. Betrachtet man nur den F_1 -Wert für die MWA-Erkennung so berichten die Autoren unter offenen Bedingungen eine F_1 -Wert von 57.24% [HSL16].

Die Autoren Cordeiro, Ramisch und Villavicencio präsentieren in der Arbeit „UFRGS&LIF at SemEval-2016 Task 10: Rule-Based MWE Identification and Predominant-Supersense Tagging“ [CRV16] ihren Ansatz für ein System mit regelbasierter MWA-Identifikation und Markierung semantischer Kategorien mithilfe von Heuristiken. Der Algorithmus zur MWA-Identifikation, der von dem System genutzt wird, basiert auf sechs verschiedenen Regelkonfigurationen, die verschiedene Klassen von MWAs betreffen. Drei dieser Regeln gelten als überwacht, da sie die bereits markierten MWAs des vorgegebenen Trainingskorpus verwenden. Jede der überwachten Regeln extrahiert dabei eine bestimmte Art von MWA aus dem Trainingskorpus, um diese als MWA-Kandidaten zu verwenden. Anschließend wird die extrahierte Menge anhand einer prozentualen Schwelle t gefiltert. Das bedeutet es werden Wortkombinationen nur dann als MWA-Kandidat ausgewählt, wenn sie in mehr als t Prozent der Fälle im Trainingskorpus als MWA markiert wurden. Die Schwellen sind nicht nach bestimmten Regeln festgelegt, sondern basieren auf den Ergebnissen, die während des Entwicklungsprozesses entstanden sind. Aus allen MWAs, die nach dieser Filterung noch vorhanden sind, wird anschließend eine Liste von MWA-Kandidaten erstellt, die dann bei der MWA-Erkennung auf einen Eingabetext abgebildet wird. Die erste Regel extrahiert alle zusammenhängenden MWAs und filtert mit einer Schwelle von 40%. Die zweite Regel extrahiert alle unzusammenhängenden MWAs und filtert mit einer Schwelle von 70%. Die dritte Regel extrahiert alle Nomen-Nomen Sequenzen aus dem Testkorpus, die mindestens einmal in dem Trainingskorpus auftreten und filtert mit einer Schwelle von 70%. Neben den genutzten überwachten Regeln, folgen noch drei unüberwachte Regeln, die nicht auf den bereits annotierten MWAs des Trainingsdatensatzes basieren, sondern weitere mögliche unmarkierte Kandidaten extrahieren. Die vierte Regel extrahiert alle Nomen-Nomen Sequenzen des Testkorpus, die nicht im Trainingskorpus auftreten. Die fünfte Regel extrahiert alle Sequenzen des POS Tags „PROP“N“. Die sechste Regel extrahiert sogenannte Verb-Partikel Kandidaten. Nachdem die MWA-Erkennung abgeschlossen ist, beginnt das System mit der Markierung von semantischen Kategorien der Wörter. Hierfür benutzt das System eine Heuristik bei der zunächst die semantischen Kategorien aus dem Trainingsdatensatz gesammelt werden. Anschließend wird jedem MWA die häufigste Kategorie, die in den Trainingsdaten gefunden wurde, zugeordnet. Einzelne Wörter werden hierbei als MWAs der Länge 1 gewertet. Die Autoren beschreiben diese Heuristik als nicht realistisch und verwenden diese nur, damit das System einen minimalen semantischen Markierer besitzt. Der eigentliche Fokus der Arbeit sollte auf der Erkennung von MWAs liegen. Das beschriebene System wurde ebenfalls bei der SemEval-2016 zur Evaluation im Bereich der geschlossenen Kategorie eingereicht. Dabei wurde ein kombinierter F_1 -Wert von 50.27%

erzielt [SHJC16]. Von den Autoren wird selbst berichtet, dass sie bei der MWA-Erkennung einen F-Wert von 51.48% erreichen [CRV16]

Ein weiterer Ansatz wird von den Autoren Björne und Salakoski in der Arbeit „UTU at SemEval-2016 Task 10: Binary Classification for Expression Detection (BCED)“ [BS16] vorgestellt. Um die zehnte Aufgabe der SemEval-2016 zu lösen, geht das System in zwei Schritten vor. Zunächst müssen die minimalen semantischen Einheiten identifiziert werden, was auch die Erkennung von MWAs einschließt. Das beschriebene System verwendet hierfür einen regelbasierten Ansatz mit einer Abbildung auf Wörterbücher. Hierfür wird versucht jedes Token eines Satzes zur längsten minimalen semantischen Einheit, zu der es gehört, zuzuordnen. Um dies umzusetzen wird jeder Satz und jedes Token Schritt für Schritt bearbeitet. Für die Überprüfung wird immer zunächst ein Token und seine fünf Nachfolgertoken als aktuelle Wortsequenz ausgewählt. Für diese Wortsequenz wird dann versucht einen Eintrag in verschiedenen Wissensquellen zu finden. Wird in keiner Wissensquelle ein Eintrag gefunden, so wird die Wortsequenz verkleinert, indem nur noch die nächsten vier Nachfolgertoken verwendet werden. Dies wird fortgeführt, bis ein Treffer gefunden wurde, oder keine Token zur Überprüfung mehr übrig sind. Sobald ein Treffer für die Sequenz von Token gefunden wird, werden diese nicht mehr betrachtet und die Suche läuft mit dem nächsten freien Token weiter. Für jeden Treffer werden ein oder mehrere Beispiele erzeugt, die später für das maschinelle Lernen verwendet werden. Hierfür erzeugen mehrere Markierungsprogramme eine Liste von möglichen semantischen Kategorien für jeden Treffer. Für jedes Paar von Tokensequenz und möglicher semantischer Kategorie wird ein Merkmalsvektor erzeugt. Anschließend wird ein binärer Klassifizierer verwendet, um diese entweder als Positiv - oder Negativbeispiele einzustufen. Die zugeordnete semantische Kategorie für jede Tokensequenz, ist schlussendlich die Kategorie, welche die höchste Wahrscheinlichkeit unter allen positiv eingestufteten Kategorien besitzt. Bei der Evaluation der zehnten Aufgabe der SemEval-2016 erreicht das System einen kombinierten F_1 -Wert von 47.13% und maximal einen F_1 -Wert von 20.11% bei der MWA-Erkennung [SHJC16]. Die Autoren berichten selbst, dass die niedrige Performanz des Systems vor allem durch die MWA-Erkennung verursacht wird.

4.2 Wortbedeutungsauflösung

Im folgenden Abschnitt werden verwandte Arbeiten vorgestellt, die sich mit den verschiedenen möglichen Ansätzen zur Wortbedeutungsauflösung befassen. Dabei ist der Abschnitt unterteilt in die wissensbasierten, überwachten und unüberwachten WSD-Systeme.

Wissensbasierte Wortbedeutungsauflösung

Im nachfolgenden wird die Funktionsweise von drei modernen wissensbasierten WSD-Systemen, die bei der Evaluation im Vergleich zu anderen Systemen einen hohen F_1 -Wert erzielten, vorgestellt. Zusätzlich dazu wird ein WSD-System erläutert, das speziell auf MWAs ausgerichtet ist.

In der Arbeit „Knowledge-based Word Sense Disambiguation using Topic Models“ [CS18] stellen die Autoren Chaplot und Salakhutdinov ihren Ansatz für die wissensbasierte Wortbedeutungsauflösung mithilfe von WordNet vor. Der beschriebene Ansatz fokussiert sich dabei auf den all-word WSD task und soll somit allen Wörtern eines Eingabetextes eine Bedeutung zuordnen. Die Menge der möglichen Bedeutungen eines Wortes wird dabei von der Wissensquelle WordNet bereitgestellt. Damit einem Wort die richtige Bedeutung zugeordnet werden kann, muss zunächst der Kontext, in dem es auftritt, betrachtet werden. Da die Sätze innerhalb eines Dokumentes nicht unabhängig voneinander sind, wird anstelle eines Satzes, das gesamte Dokument als Kontext für die Bedeutungsauflösung

eines Wortes verwendet. Hierfür wird eine abgewandelte Version des Themenmodells „Latent Dirichlet Allocation“ (LDA) verwendet. LDA verwendet normalerweise eine Menge von verschiedenen Themen, um Dokumente zu repräsentieren. Die abgewandelte Version nimmt hierfür stattdessen die Wahrscheinlichkeit verschiedener Synsets. Die eigentliche Bedeutungsauflösung erfolgt dann mit den Definitionen aus WordNet. Zur Evaluation des beschriebenen Systems wurden die Benchmarks aus den SensEval-2, SensEval-3, SemEval-2013 und SemEval-2013 Workshops für die Berechnung des F_1 -Wertes verwendet. Das System erreicht seinen maximalen F_1 -Wert bei der SemEval-2015 mit 69.6%.

In der Arbeit „Personalizing PageRank for Word Sense Disambiguation“ [AS09] stellen die Autoren Agirre und Soroa erstmals ihre graphbasierte Methode für wissensbasierte WSD-Systeme vor. Da lexikalische Wissensbasen durch eine Menge von Konzepten und deren Beziehungen zueinander geformt werden, lässt sich ein Graph extrahieren bei dem die Knoten die Konzepte der Wissensbasis und die Kanten die Beziehungen zwischen diesen Konzepten darstellen. Die Autoren beschreiben zunächst, dass ein PageRank Algorithmus verwendet wird, welcher jedem Knoten einen Rang basierend auf seiner strukturellen Bedeutung innerhalb des Graphen zuweist. Der Rang kann als Wahrscheinlichkeit dafür betrachtet werden, dass ein zufälliger Lauf über den Graphen in einem genügend großen Zeitfenster in dem entsprechenden Knoten endet. Dabei ist die Übergangswahrscheinlichkeit für alle Nachbarknoten eines Knoten v gleich. Um diesen Algorithmus für die Wortbedeutungsauflösung zu verwenden, benutzen die Autoren den WordNet Graphen und die Beziehungen zwischen dessen Konzepten. Der traditionelle PageRank Algorithmus hat das Problem, dass eine kontextunabhängige Rangfolge erstellt wird. Bei der Wortbedeutungsauflösung soll jedoch der Eingabekontext bei der Auswahl der passenden Wortbedeutungen einbezogen werden. Das System besitzt zwei unterschiedliche Ansätze, um dieses Problem zu lösen. Der erste Ansatz erzeugt anhand des Eingabetextes zunächst einen Disambiguierungsgraph, indem eine Breitensuche im unterliegenden Graph der Wissensbasis für alle Konzepte, die zu den Wörtern des Eingabekontextes gehören, durchgeführt wird. Jede Durchführung berechnet den kürzesten Weg zwischen einem Konzept v und den restlichen Konzepten des Graphen. Dies wird für alle Worte des Eingabekontexts wiederholt. Der Disambiguierungsgraph ergibt sich dann als Vereinigung der Knoten und Kanten der kürzesten Wege. Anschließend wird der traditionelle PageRank Algorithmus auf dem Disambiguierungsgraphen durchgeführt und jedem Wort die Bedeutung mit maximalem Rang zugeordnet. Der zweite Ansatz verwendet den gesamten Graphen der Wissensbasis und wendet dabei den personalisierten PageRank Algorithmus an, welcher eine abgewandelte Version des traditionellen PageRank Algorithmus darstellt. Hierbei werden die Wörter des Eingabekontextes direkt in den Graphen eingefügt und mit ihren entsprechenden Konzepten durch Kanten verbunden. Im Gegensatz zu dem traditionellen PageRank Algorithmus, sind bei dem personalisierten PageRank Algorithmus nicht alle Übergänge gleich wahrscheinlich. Bei der Berechnung des personalisierten PageRanks erhalten die hinzugefügten Knoten eine erhöhte Wahrscheinlichkeit. Alle Konzepte, mit denen die Wörter im Kontext verbunden sind, erhalten somit eine höhere Relevanz, wodurch automatisch der Kontext bei der Bedeutungsauflösung einbezogen wird. Der berechnete PageRank kann anschließend als Messwert für die strukturelle Relevanz eines Konzeptes in Abhängigkeit vom Eingabekontext gesehen werden. Auch hier wird jedem Wort dann das Konzept mit dem höchsten Rang als Bedeutung zugeordnet. Bei der Evaluation des Systems in der Arbeit „Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison“ [RCCN17] wurden die Datensätze der Senseval-2, Senseval-3, SemEval-07, SemEval-13 und SemEval-15 verwendet. Bei der Evaluation mit dem Datensatz der SemEval-15 erreichte das System mit einem F_1 -Wert von 61.2% sein Maximum.

Die Autoren Agirre, Lacalle und Soroa zeigen in der Arbeit „The risk of sub-optimal use of Open Source NLP:Software UKB is inadvertently state-of-the-art in knowledge-

based WSD“ [ALS18] jedoch, dass das zuvor beschriebene System UKB, bisher aufgrund schlecht gewählter Parameter suboptimale Ergebnisse bei den Evaluationen erzielte. Obwohl die optimalen Parameter in den wissenschaftlichen Artikeln genannt wurden, wurde die Standardversion der Programmsammlung nicht mit diesen veröffentlicht, was zu verfälschten Ergebnissen führte. Für die Evaluation des Systems mit optimalen Parametern wurden dieselben Datensätze wie in der Arbeit von Raganato, Camacho-Collados und Navigli [RCCN17] verwendet. Bei der Aufgabe der SemEval-2015 erreichte UKB einen F_1 -Wert von 70.3%, was eine deutliche Verbesserung gegenüber den zuvor berichteten 61.2% darstellt.

In der Arbeit „Entity Linking meets Word Sense Disambiguation: a Unified Approach“ [MRN14] beschreiben die Autoren Moro, Raganato und Navigli einen weiteren graphbasierten Ansatz zur wissensbasierten Wortbedeutungsauflösung. Sie unterscheiden dabei zwischen Entitätenverknüpfung und Wortbedeutungsauflösung, wobei bei der Entitätenverknüpfung versucht wird Entitäten innerhalb eines Textes zu den passenden Einträgen in einer Wissensbasis zuzuordnen. Die Wortbedeutungsauflösung ist der Entitätenverknüpfung sehr ähnlich, jedoch beschränkt diese sich auf die auf Entitäten, deren voller Name in der Wissensquelle zu finden ist. Die Entitätenverknüpfung kann jedoch auch Teilworte mit Konzepten verknüpfen. Da beide Aufgaben eng miteinander verbunden sind schlagen die Autoren einen uniformen Ansatz zu beiden Aufgaben vor. Generell läuft der Prozess in drei Schritten ab. Zunächst wird jedem Knoten eines gegebenen semantischen Netzwerks eine semantische Signatur, die aus einer Menge von verwandten Knoten besteht, zugewiesen. Das verwendete semantische Netzwerk ist hierbei Babelnet 1.1.1, da es die größte multilinguale Wissensbasis ist und sowohl WordNet, als auch Wikipedia verwendet. Zur Bestimmung der semantischen Signaturen wird den Kanten des Netzwerks, die in dicht verknüpften Bereiche liegen, ein höheres Gewicht zugewiesen. Anschließend wird für jedes Konzept und jede Entität des Netzwerks ein „random walk with restart“ durchgeführt. Die semantische Signatur eines Konzeptes oder einer Entität ergibt sich dann aus der Menge an Knoten, deren Anzahl an Berührungen während des Verfahrens eine bestimmte Grenze überschreitet. Anschließend werden in einem gegebenen Text, alle Bestandteile, die mit dem Netzwerk verknüpfbar sind, extrahiert und deren mögliche Bedeutungen anhand des Netzwerks aufgelistet. Innerhalb eines Ausdrucks können dabei mehrere verknüpfbare Bestandteile vorhanden sein, die alle disambiguiert werden. Schlussendlich wird eine graphbasierte semantische Interpretation des Eingabetextes erzeugt, indem die möglichen Bedeutungen der extrahierten Bestandteile anhand der semantischen Signaturen miteinander verknüpft werden. Aus dieser Repräsentation wird dann ein dichter Subgraph extrahiert, aus dem dann die besten Bedeutungen für jeden Bestandteil des Textes entnommen werden. Genau wie UKB, wurde auch Babelfy in der Arbeit „Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison“ [RCCN17] evaluiert. Bei der Evaluation mit dem Datensatz der SemEval-15 erreicht Babelfy mit dem F_1 -Wert 70.3% sein Maximum.

In der Arbeit „Multiwords and Word Sense Disambiguation“ [AAC05] stellen die Autoren Arranz, Atserias und Castillo ein wissensbasiertes WSD-System vor, welches speziell die Aufgabe besitzt, MWAs zu erkennen und mithilfe von WordNet eine Bedeutungsauflösung durchzuführen. Das beschriebene WSD-System verwendet ein sequentielles Model mit verschiedenen Stufen. Dabei erhält jede Stufe die verarbeiteten Informationen der vorangehenden Stufe und verwendet diese weiter. Bevor das System die Wortbedeutungsauflösung bei einem Eingabetext durchführt, werden zunächst drei Verarbeitungsstufen durchlaufen. In der ersten Stufe wird der Text zunächst tokenisiert, anschließend werden die Tokens mit POS-Markierungen versehen und mithilfe von WordNet lemmatisiert. Die zweite Stufe ist für die Erkennung aller möglichen MWAs innerhalb des Textes zuständig. Hierfür werden zunächst aufeinanderfolgende Token innerhalb eines Satzes mit einer ge-

schlossenen Liste von MWAs aus WordNet verglichen. Ein Problem, das bei dieser Technik auftritt, sind die verschiedenen möglichen morphologischen Inflektionen eines MWAs, die nicht in der WordNet-Liste repräsentiert werden. Deshalb untersuchte ein Sprachexperte zunächst welche morphologischen Inflektionen bei bestimmten syntaktischen Mustern der MWAs auftreten können, indem er die MWAs des SemCor-Korpus analysierte. Diese syntaktischen Muster wurden anschließend verwendet, um die Anzahl der zu untersuchenden Variationen eines MWAs zu verringern und anschließend eine Liste aller möglichen MWAs zu erzeugen. Nach der MWA-Detektion folgt die dritte Stufe, die entscheiden soll, ob ein erkannter MWA auch wirklich als solcher im Satz agiert. Dies wird als MWA-Selektion bezeichnet. Hierfür wurden zwei mögliche Heuristiken getestet. Die erste Heuristik erkennt immer den längsten MWA aus der Liste als passend an. Für die zweite Heuristik wurde erneut der SemCor-Korpus verwendet, um zu untersuchen, ob die Wörter eines möglichen MWA häufiger isoliert oder wirklich als echter MWA auftreten. Anhand dieser Statistik können dann bestimmte MWAs ausgeschlossen oder ausgewählt werden. Anschließend folgt der letzte Schritt, bei dem das System eine Wortbedeutungsauflösung mithilfe von Heuristiken, die auf den Funktionen der Wissensquelle WordNet basieren, durchführt. Für die Evaluation wurde eine Menge von Definitionen aus dem erweiterten WordNet 2.0 verwendet, deren Wörter alle manuell mit einer Bedeutung annotiert wurden. Während der Evaluation wurden beide Heuristiken für die MWA-Selektion evaluiert. Im besten Fall erreichte das System einen F_1 -Wert von 0.81.

Überwachte Wortbedeutungsauflösung

In diesem Abschnitt werden zwei Systeme für die überwachte Wortbedeutungsauflösung, welche einen hohen F_1 -Wert bei der Evaluation erzielten, näher beschrieben.

In der Arbeit „SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation“ [PRDB17] beschreiben die Autoren Papandrea, Raganato, und Bovi die von ihnen entwickelte Java API SupWSD für die überwachte Wortbedeutungsauflösung. Zunächst wird der Eingabetext vom Eingabeparser eingelesen, in verschiedene Satzgruppen eingeteilt und anschließend von einer Textvorverarbeitungspipeline bearbeitet. Der vorverarbeitete Text wird anschließend an das Modul für die Merkmalsextraktion weitergeleitet, wo anschließend verschiedene Merkmale, die später für das Training des überwachten Modells verwendet werden, konstruiert werden. SupWSD erlaubt die Konfiguration der zu verwenden Merkmale mithilfe einer Konfigurationsdatei, jedoch werden standardmäßig der POS Tag, die Token in der Umgebung eines Zielwortes, die geordneten Sequenzen von Token in der Umgebung eines Zielwortes, vortrainierte Worteinbettungen und die syntaktischen Beziehungen basierend auf dem Abhängigkeitsbaum eines Satzes als Merkmale verwendet. Schlussendlich werden die konstruierten Merkmale im letzten Modul dazu verwendet einen Klassifikator zu trainieren. Bei der Evaluation wurden die zwei Textkorpora SemCor und OMSTI verwendet, um das System zu trainieren und anschließend auf mehrere Aufgaben anzuwenden. Beispielsweise erreichte SupWSD nach dem Training auf dem SemCor-Korpus bei der SemEval2015 Aufgabe einen F_1 -Wert von 70.0%.

Die Autoren Huang et al. beschreiben in der Arbeit „GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge“ [HSQH19] ihr überwachtes System für die Wortbedeutungsauflösung. Obwohl überwachte WSD-Systeme generell bessere Ergebnisse erzielen, als wissensbasierte Ansätze, mangelt es ihnen dennoch an Flexibilität. Das beschriebene System versucht deshalb die Wortbedeutungen aus WordNet in den Prozess der überwachten Wortbedeutungsauflösung einzubeziehen. Hierfür verwendet das System das neue Modell für Sprachrepräsentation BERT, das bereits auf einem großen Textkorpus vortrainiert wurde. BERT ist darauf trainiert bidirektionale Repräsentationen durch unmarkierte Textkorpora zu lernen [DCLT19]. Damit ist es BERT möglich die Beziehung zwischen Paaren von Sätzen explizit zu modellieren, weshalb die Autoren versuchen

die Wortbedeutungsauflösung, als ein Satzpaarklassifikationsproblem zu formulieren. Dazu werden zunächst alle vorhandenen Definitionen eines Zielwortes aus der Wissensquelle WordNet extrahiert. Anschließend wird aus jeder dieser Definitionen und dem Satz, in dem das Zielwort auftritt, ein Kontext-Wortbedeutungspaar gebildet, welche schlussendlich als Eingabe für das BERT-Modell verwendet werden. Damit das BERT-Modell nun eine passende Ausgabe für die Wortbedeutungsauflösung erstellt, wird eine zusätzliche Klassifikationsschicht hinzugefügt, die jedem Kontext-Wortbedeutungspaar ein Label „ja“ oder „nein“ anhängt. Bei der Evaluation mit dem SemEval-2015 Datensatz erreichten sie einen maximalen F_1 -Wert von 80.4%.

Unüberwachte Wortbedeutungsauflösung

In diesem Abschnitt wird ein System für die unüberwachte Wortbedeutungsauflösung näher beschrieben. Wie bereits in Kapitel 2 beschrieben wurde, unterscheidet sich die Funktionsweise von unüberwachten Systemen von den wissensbasierten und überwachten Systemen, da diese nicht mit bereits vorhandenen Wortdefinitionen arbeiten.

Die Autoren Pelevina et al. stellen in der Arbeit „Making Sense of Word Embeddings“ [PABP17] ihren Ansatz für die unüberwachte Wortbedeutungsauflösung mithilfe von bereits vorhandenen Worteinbettungen vor. Es wird beschrieben, dass die meisten Systeme Worteinbettungen entweder direkt durch Textkorpora lernen oder hierfür ein bereits vorhandenes Inventar einer lexikalischen Ressource verwenden. Das beschriebene System soll jedoch anhand von vorhandenen Worteinbettungen, selbst ein Bedeutungsinventar erzeugen, welches anschließend für die Wortbedeutungsauflösung verwendet wird. Zunächst wird das System mit Worteinbettungen, die 100 bis 300 Dimensionen besitzen, trainiert. Anschließend werden für die Wörter Graphen basierend auf der Ähnlichkeit von Wörtern erzeugt, indem für jedes Wort die 200 nächsten Nachbarn extrahiert werden. Im nächsten Schritt wird ein Bedeutungsinventar erzeugt, das Wortbedeutungen in Form von Wortclustern repräsentiert. Hierfür wird auf den zuvor erstellten Graphen ein Algorithmus angewendet, der Cluster erzeugt und anschließend werden diese als Wortbedeutungen verwendet. Die gefundenen Wortbedeutungen und die bereits vorhandenen Wortvektoren werden dann für die Berechnung von Bedeutungsvektoren genutzt. Schlussendlich wird bei der Wortbedeutungsauflösung ein Wort zunächst auf seine möglichen Bedeutungsvektoren abgebildet und anschließend werden zwei mögliche Strategien verwendet, um eine korrekte Wortbedeutung auszuwählen. Die erste Strategie basiert dabei auf der Bedeutungswahrscheinlichkeit in einem gegebenen Kontext, während die zweite Strategie auf der Ähnlichkeit zwischen Bedeutung und Kontext basiert.

In der Arbeit „Unsupervised, Knowledge-Free, and Interpretable Word Sense Disambiguation“ [PMR⁺17] beschreiben Pachenko et al. ein System für die unüberwachte und wissensfreie Wortbedeutungsauflösung. Sie beschreiben dabei, dass die Lösungen wissensbasierter WSD-Systeme besser interpretierbar sind, da sie Zugang zu einer großen Menge manuell kodierter Informationen besitzen. Diese Informationen müssen jedoch unter hohem manuellen Aufwand selbst erstellt werden. Im Gegensatz dazu erfordern unüberwachte Systeme keinen direkten manuellen Aufwand, was jedoch zu einer geringeren Interpretierbarkeit führt. Das beschriebene System geht dabei zunächst so vor, dass aus verschiedenen Textkorpora zunächst ein sogenanntes WSD-Modell erzeugt wird, welches später für die Wortbedeutungsauflösung verwendet wird. Hierfür wird zunächst ein Graph aus semantisch verwandten Wörtern extrahiert. In diesem Graph werden anschließend Cluster von Wörtern gebildet, um eine Wortbedeutungsinduktion durchzuführen. Jede Wortbedeutung wird dabei als ein Wortcluster dargestellt. Für die zugehörigen Wörter eines Clusters werden außerdem kontextuelle Hinweise gesammelt, um eine korrekte Wortbedeutungsrepräsentation zu ermöglichen. Damit die Interpretierbarkeit dieser Darstellung erweitert wird, werden aus dem Korpus Hyperonyme extrahiert, die anschließend zu den

passenden Clustern zugeordnet werden. Die beschriebenen Schritte bilden das sogenannte WSD-Modell. Dies wird nun verwendet, um eine Liste von Beispielsätzen, für die entsprechenden Wortcluster, zu extrahieren. Die erzeugte Wissensbasis wird nun verwendet, um eine Wortbedeutungsauflösung bei einzelnen, oder bei allen Wörtern eines Satzes durchzuführen. Hierfür wird der Kontext der Eingabe in Betracht gezogen und anschließend eine Liste der wahrscheinlichsten Bedeutungen ausgegeben. Panchenko et al. [PRF⁺17] berichten in ihrer Evaluation des Systems vergleichbare Ergebnisse zu anderen modernen unüberwachten Systemen. Diese weisen jedoch nicht die gleiche Interpretierbarkeit wie das präsentierte System auf.

5 Analyse und Entwurf

Wie bereits in Kapitel 3 beschrieben, versucht das Projekt INDIRECT die direkte Verknüpfung zwischen Anforderungen und Quelltext innerhalb eines Softwareprojektes zu nutzen, um die Wartung und Pflege des Codes zu verbessern. Hierfür werden Anforderungen und Quelltext analysiert, in Form von Graphen modelliert und anschließend werden diese Modelle verwendet, um automatisch wechselseitige Rückverfolgbarkeitsinformationen zu erzeugen.

Damit ein System ein Dokument in natürlicher Sprache automatisch verarbeiten kann, muss dieses System die einzelnen Bedeutungen der vorkommenden Wörter verstehen. Der Prozess, einem Wort seine geeignete Bedeutung innerhalb eines Kontextes zuzuordnen, nennt sich Bedeutungsauflösung. Eine solche Bedeutungszuordnung ist zwar manuell möglich, jedoch steigt der erforderliche Aufwand hierfür mit der Anzahl der formulierten Anforderungen enorm an. Es existieren deshalb bereits Ansätze für die automatische Wortbedeutungsauflösung einzelner Wörter. In einer Sprache existieren jedoch Ausdrücke, die zwar aus mehreren Wörtern zusammengesetzt sind, jedoch als Einheit eine einzelne Bedeutung tragen. Wie man bereits anhand von Beispiel 1.1 gesehen hat, reicht eine Bedeutungsauflösung der einzelnen Teilworte nicht immer aus, um die korrekte Bedeutung des gesamten Ausdrucks zu erfassen. In dieser Arbeit werden solche Ausdrücke als Multiwort-Ausdrücke (MWAs) bezeichnet. Vor allem im Bereich der Anforderungsdokumente werden häufig Fachbegriffe und Eigennamen verwendet, die also solche MWAs agieren. Eine Bedeutungsauflösung auf den einzelnen Wörtern ist also für das Projekt INDIRECT in einigen Fällen nicht ausreichend.

5.1 Ziele dieser Arbeit

Das Ziel dieser Arbeit ist die Entwicklung eines WSD-Systems, dem es möglich ist, eine Bedeutungsauflösung mit Berücksichtigung der oben erwähnten MWAs durchzuführen. Wichtig ist hierbei zu erwähnen, dass das System vor allem mit Anforderungsdokumenten arbeitet, weshalb das System auch eine Bedeutungsauflösung bei domänenspezifischen Fachbegriffen durchführen können sollte. Die zu erfüllende Aufgabe wird anhand von drei Teilzielen, die erreicht werden müssen, genauer gegliedert.

1. Das erste Teilziel dieser Arbeit besteht darin den Begriff „Multiwort-Ausdruck (MWA)“ zu definieren und abzugrenzen. Diese Definition wird anschließend in den zwei folgenden Teilzielen verwendet.

2. Anschließend soll eine verlässliche Methode entwickelt werden MWAs innerhalb eines Anforderungsdokumentes zu erkennen und diese als kleinste semantische Einheit aufzufassen.
3. Nachdem die MWAs in den Anforderungsdokumenten identifiziert wurden, soll eine Methode zur Bedeutungsauflösung mit Berücksichtigung von MWAs entwickelt werden.

5.2 Definition des Begriffs „Multiwort-Ausdruck“

Im nachfolgenden Abschnitt wird der Begriff „Multiwort-Ausdruck“ näher definiert und abgegrenzt. Dabei wird vor allem auf die verschiedenen Arten von MWAs und welche davon im Kontext dieser Arbeit relevant sind eingegangen. In dem Artikel „Multiword Expressions: A Pain in the Neck for NLP“ [SBB⁺02] schlagen die Autoren Sag et al. eine Definition von Multiwort-Ausdrücken auf Basis ihrer lexikalischen Veränderbarkeit vor, die in der nachfolgenden Analyse ebenfalls verwendet wird.

5.2.1 Lexikalisierte Phrasen

Als lexikalisierte Phrasen bezeichnen die Autoren Ausdrücke, die zumindest teilweise semantisch oder syntaktisch idiosynkratisch sind. Das bedeutet, dass diese Ausdrücke bestimmte Eigenschaften besitzen, die sich nicht anhand von allgemein bekannten Regeln ableiten lassen. Ein Beispiel hierfür wären Redewendungen wie „spill the beans“ [SBB⁺02] deren Bedeutung, nicht durch die Bedeutung der einzelnen Teilworte ableitbar sind. Die lexikalisierten Phrasen können anhand ihrer lexikalischen Veränderbarkeit in weitere Unterkategorien aufgeteilt werden.

Feste Ausdrücke

Die festen Ausdrücke sind eine Menge von Ausdrücken, die vollständig unveränderbar sind und somit nicht den grammatischen Konventionen folgen.

Beispiel 5.1: Beispiele für feste Ausdrücke [SBB⁺02]

- by and large
- in short
- every which way

Das Beispiel Beispiel 5.1 zeigt einige feste Ausdrücke, die sich in keiner Art und Weise verändern lassen. Der Ausdruck „by and large“ hat beispielsweise eine feste Bedeutung und durch Veränderungen wie „by and very large“, oder „by and larger“, geht die eigentliche Bedeutung des Ausdrucks verloren.

Semi-feste Ausdrücke

Semi-feste Ausdrücke unterliegen, ähnlich wie die festen Ausdrücke, ebenfalls einigen Einschränkungen bei der Veränderbarkeit, jedoch sind lexikalische Variationen bis zu einem bestimmten Grad erlaubt. Die Autoren unterscheiden in diesem Fall die nicht-zerlegebaren Redewendungen, zusammengesetzte Nomen und Eigennamen. Bei der Zerlegbarkeit einer Redewendung beziehen sich die Autoren in diesem Fall auf die semantische Zerlegbarkeit. Ein Beispiel hierfür ist die oben erwähnte Redewendung „spill the beans“, die äquivalent ist zu dem Ausdruck „ein Geheimnis verraten“. Sie lässt sich zerlegen in die Bestandteile

Tabelle 5.1: Beispiele für semi-feste Ausdrücke [SBB⁺02]

Art des Ausdrucks	Ausdruck	erlaubte Variation
nicht-zerlegbare Redewendung	kick the bucket	kicked the bucket
nicht-zerlegbare Redewendung	wet himself	wet themselves
zusammengesetztes Nomen	car park	car parks
Eigennamen	San Francisco 49ers	49ers

„spill“, also das Verraten und „the beans“, also das Geheimnis. In nicht-zerlegbaren Redewendungen ist solch eine semantische Aufteilung nicht möglich. In Tabelle 5.1 sind einige Beispiele für semi-feste Ausdrücke und ihre möglichen Variationen zu sehen. Die Autoren beschreiben, dass nicht-zerlegbare Redewendungen lediglich Inflektionen wie „kicked the bucket“ und Variationen der Reflexivpronomen wie „wet themselves“ erlauben. Bei zusammengesetzten Nomen ist lediglich der Wechsel von der Singular - zur Pluralform, durch das Anhängen eines „s“ möglich. Bei den Eigennamen werden keine festen Regeln durch die Autoren festgelegt.

Syntaktische flexible Ausdrücke

Syntaktisch flexible Ausdrücke erlauben, im Gegensatz zu festen und semi-festen Ausdrücken, eine deutliche größere Breite an syntaktischen Variationen. Im Gegensatz zu den semi-festen und festen Ausdrücken, ist eine genaue Festlegung, welche lexikalischen Variationen erlaubt sind, nicht möglich. Die Autoren unterscheiden bei den syntaktisch flexiblen Ausdrücken zwischen Verb-Partikel Konstruktionen, zerlegbaren Redewendungen und Funktionsverbkonstrukten. Verb-Partikel Konstruktionen bestehen dabei aus einem Verb gefolgt von einem oder mehr Partikeln. Es besteht außerdem die Möglichkeit, dass zwischen Verb und Partikel eine Nominalphrase steht. Zerlegbare Redewendungen wurden bereits in Absatz 5.2.1 in Bezug auf die nicht-zerlegbaren Redewendungen definiert. Die

Tabelle 5.2: Beispiele für semi-feste Ausdrücke [SBB⁺02]

Art des Ausdrucks	Ausdruck
Verb-Partikel Konstruktion ohne Nominalphrase	look up
Verb-Partikel Konstruktion mit Nominalphrase	call Kim up
zerlegbare Redewendung	spill the beans
Funktionsverbkonstrukt	make a mistake

Tabelle 5.2 listet einige Beispiele für die verschiedenen Arten von syntaktisch flexiblen Ausdrücken auf. Verb-Partikel Konstruktionen wie der Ausdruck „call up“ erlauben beispielsweise sowohl die Form „call Kim up“, als auch „call up Kim“. Die semantische Zerlegbarkeit der Redewendung „spill the beans“ wurde bereits in Absatz 5.2.1 näher erläutert.

5.3 MWAs im Kontext dieser Arbeit

Der Begriff „Multiwort-Ausdruck“, der im Kontext dieser Arbeit verwendet wird, bezieht sich lediglich auf einen Ausschnitt der in der Arbeit „Multiword Expressions: A Pain in the Neck for NLP“ [SBB⁺02] definierten MWAs. Das zu entwickelnde WSD-System arbeitet vorwiegend auf Anforderungsspezifikationen, die aufgrund von Konventionen, häufig kurze Sätze und wenig komplexe Wortkonstrukte beinhalten. Vor allem Ausdrücke wie zerlegbare und nicht-zerlegbare Redewendungen spielen bei Anforderungen für Softwareprojekte eine geringe Rolle, weshalb die Erkennung dieser Ausdrücke nur wenig Relevanz

besitzt. Wie bereits in Kapitel 3 beschrieben, versucht INDIRECT in seinem Anforderungsmodell vor allem Informationen bezüglich Entitäten und der Beziehungen zwischen diesen Entitäten in Form eines Graphen zu modellieren. Der Fokus dieser Arbeit sollte also darauf liegen, solche Entitäten als MWA zu erkennen und ihnen eine Bedeutung zuzuordnen. Aus diesem Grund werden die Verb-Partikel Konstruktionen, so wie die Funktionsverbkonstrukte ebenfalls nicht in den Begriff „Multiwort-Ausdruck“ einbezogen. Bei der Erkennung der für INDIRECT relevanten Entitäten spielen jedoch zusammengesetzte Nomen und Eigennamen eine große Rolle. Beispiele für zusammengesetzte Nomen wie „computer network“, „database management system“, „system administrator“, oder auch Eigennamen wie „Microsoft Word“, „Mozilla Firefox“, geben wichtige Hinweise bezüglich der Nutzung bestimmter Werkzeuge, welche Personen Teile des Systems verwenden, oder auch welche Teile des Systems miteinander in Verbindung stehen. Insgesamt bezieht sich der Begriff „Multiwort-Ausdruck“ im Kontext dieser Arbeit auf die zusammengesetzten Nomen und die Eigennamen innerhalb von Anforderungsspezifikationen.

Nachdem nun der Begriff Multiwort-Ausdruck abgegrenzt wurde, wird ein besonderer Randfall betrachtet, der vor allem bei den Nomen auftreten kann.

Beispiel 5.2: Adjektiv+Nomen Kombination als MWA

The system calculates the **expected value**

Kombinationen von Adjektiven und deren Nomen wie beispielsweise „large car“, „green wall“, „high tower“, etc. besitzen keine Bedeutung, die nicht anhand der einzelnen Teilbedeutungen der Worte erkannt werden kann. In diesem Fall reicht eine einfache Wortbedeutungsauflösung der einzelnen Wörter aus, um die Aussage zu verstehen. Beispiel 5.2 zeigt jedoch einen Fall bei der eine Adjektiv und Nomen Kombination in Form eines MWAs auftritt. Der Ausdruck „expected value“ entspricht dem Erwartungswert in der Wahrscheinlichkeitstheorie und besitzt somit eine besondere Bedeutung, die sich nicht anhand der Teilbedeutungen ableiten lässt. Solche besonderen Fälle von Adjektiv und Nomen Kombinationen, werden deshalb ebenfalls in den Begriff „Multiwort-Ausdruck“ aufgenommen.

5.4 Erkennung der MWAs

Die Identifikation der MWAs stellt das zweite Teilziel dieser Arbeit dar und wird im nachfolgenden Abschnitt näher analysiert. Wie bereits in Abschnitt 5.2 lässt sich die genaue Bedeutung eines MWAs in einigen Fällen nicht anhand der Bedeutungen der einzelnen Teilworte ableiten.

Beispiel 5.3:

1. „He eats a **piece of cake**.“
2. „This test was a **piece of cake**.“

Das Beispiel 5.3 zeigt die zwei unterschiedlichen Bedeutungen des Ausdrucks „a piece of cake“. Im ersten Satz ist es theoretisch möglich, jedem Wort des Ausdrucks eine eigenständige Bedeutung zuzuordnen, und dennoch den eigentlichen Sinn des gesamten Ausdrucks richtig zu interpretieren. Im zweiten Satz hingegen repräsentiert der Ausdruck „a piece of cake“ eine Redewendung, die ausdrücken soll, dass eine Handlung einfach durchzuführen ist. Eine Bedeutungsauflösung bei den einzelnen Teilworten macht in diesem Fall also

keinen Sinn, da die eigentliche Bedeutung des Ausdrucks nicht von diesen abhängig ist. Damit ein WSD-System solche Sätze dennoch möglichst genau interpretieren kann, ist es zunächst wichtig zu erkennen, welche Worte eines Satzes zusammen eine minimale semantische Einheit bilden. Die erste Aufgabe dieser Arbeit besteht also darin, eine Methode zu erarbeiten, die es ermöglicht, MWAs innerhalb eines Textes zu erkennen.

5.4.1 MWA-Erkennung anhand vorgefertigter Listen

Ein Ansatz, der versucht das Problem auf eine simple Art und Weise zu lösen, ist die MWA-Erkennung anhand von vorgefertigten Listen. Das Grundprinzip ist hierbei, zunächst eine möglichst große Liste von MWAs anzufertigen und anschließend im zu verarbeitenden Text nach Wortkombinationen, die sich in der Liste befinden zu suchen, um mögliche MWA-Kandidaten zu identifizieren. Da bei diesem Ansatz die Liste die einzige Informationsquelle für die Identifikation darstellt, hat ihre Größe einen maßgeblichen Einfluss auf die Anzahl der Kandidaten, die identifiziert werden können. Außerdem muss entschieden werden, ob eine Wortkombination der Eingabe wirklich einem MWA der Liste entspricht, oder nur eine Sequenz von isolierten Wörtern darstellt, die zufällig mit einem MWA in der Liste übereinstimmt. Wie in Beispiel 5.3 bereits gezeigt, besteht nämlich die Möglichkeit, dass eine Wortkombination zwar in der Liste vorkommt, jedoch in dem Satz nicht die Funktion eines MWAs übernimmt.

Die Arbeit „UTU at SemEval-2016 Task10: Binary Classification for Expression Detection (BCED)“ [BS16] verwendet eine mögliche Implementierung des oben beschriebenen Ansatzes für die MWA-Erkennung. Um eine möglichst große Anzahl an möglichen MWA-Kandidaten zu identifizieren, wird eine Menge von Markierungsprogrammen verwendet, die ihre Informationen aus verschiedenen Wissensquellen beziehen. Um ein MWA innerhalb eines Satzes zu identifizieren wird eine Sequenz bestehend aus einem Token und seinen fünf Nachfolgertoken ausgewählt und anschließend wird diese Tokenkombination mit den verschiedenen Wissensquellen abgeglichen. Wird kein Eintrag gefunden, so wird mit einer kürzeren Sequenz weitergesucht, indem nur die nächsten vier Nachfolgertoken verwendet werden. Dies wird fortgeführt bis ein Eintrag gefunden wird oder in der Sequenz keine Token mehr übrig sind. Anschließend wird die Suche beim nächsten freien Token fortgeführt. Im Endeffekt erkennt das System immer den ersten und auch längsten MWA-Kandidaten ausgehend vom Satzanfang. Es wird nicht in Betracht gezogen, dass möglicherweise ein kürzerer MWA-Kandidat vorhanden ist, dessen Bedeutung passender wäre und außerdem werden nicht zusammenhängende MWAs vollständig von diesem Ansatz ignoriert.

Beispiel 5.4:

„He passed out of sight.“ [AAC05]

Beispiel 5.4 zeigt einen möglichen Fall, bei dem die Methode einen inkorrekten MWA-Kandidaten auswählen könnte. Nimmt man an, dass der gesamte Ausdruck „passed out of sight“, nicht in einer Wissensquelle vorhanden ist, aber der Ausdruck „passed out“ ist vorhanden, so würde die Methode den Ausdruck „passed out“ als MWA erkennen. In dem oben genannten Beispiel wäre jedoch der Ausdruck „out of sight“ ein besserer Kandidat. Bei der SemEval-2016 [SHJC16] erreicht das System bei der Identifikation von MWAs in TED talks mit einem Wert von 20.11% seinen höchsten F₁-Wert und schneidet somit deutlich schlechter ab, als die anderen Systeme. Eine Nutzung des Systems wird deshalb in dieser Arbeit nicht in Betracht gezogen.

In der Arbeit „Multiwords and Word Sense Disambiguation“ [AAC05] wird ebenfalls ein listenbasierter Ansatz zur MWA-Erkennung vorgestellt. Das beschriebene System arbei-

tet für die Identifikation von MWA-Kandidaten mit einer Liste von möglichen MWAs aus WordNet. Für die Auswahl passender Kandidaten testen die Autoren ebenfalls die Heuristik, die immer den längsten MWA auswählt. Zusätzlich verwenden sie noch eine Heuristik basierend auf Statistiken. Hierfür wurde das SemCor-Korpus untersucht, um herauszufinden, ob ein MWA-Kandidat häufiger als echter MWA oder als eine Sequenz von einzelnen Wörtern auftritt. Basierend auf dieser Untersuchung, wird anschließend ein MWA-Kandidat verworfen, oder ausgewählt. Bei der Evaluation erzielte diese Heuristik bessere Ergebnisse bei der Präzision. Man muss jedoch beachten, dass hierfür eine große Menge an Daten nötig ist, damit eine relevante Aussage getroffen werden kann. Insgesamt erreicht das System bei der MWA-Erkennung einen F_1 -Wert von 0.81. Obwohl die Autoren ebenfalls einen listenbasierten Ansatz verwenden, ist der Wert deutlich besser, als bei der Arbeit „UTU at SemEval-2016 Task10: Binary Classification for Expression Detection (BCED)“ [BS16]. Es liegt also die Vermutung nahe, dass dies lediglich an dem verwendeten Testdatensatz liegt. Wie bereits zuvor erwähnt, ist die Menge der in der Liste vorhandenen MWAs für listenbasierte Ansätze wichtig. Die Autoren beziehen ihre Liste von MWA-Kandidaten aus der Wissensquelle WordNet, führen dabei aber auch gleichzeitig ihre Evaluation auf den WordNet Glossen durch. Es liegt somit die Vermutung nahe, dass der hohe Wert, durch die Überschneidung der MWAs in den Glossen mit den MWAs in der WordNet Liste entstanden ist. Eine Einschätzung der Performanz des Systems auf beliebigen Eingabetexten ist somit nicht möglich.

Ein ebenfalls wichtiger Punkt, der bei der regelbasierten MWA-Identifikation mit Listen beachtet werden muss, ist die morphologische Inflektion. Wörter einer Sprache verändern ihre Form anhand des grammatischen Kontexts, in dem sie auftreten. Beispielsweise werden Verben in Abhängigkeit von der ausführenden Person konjugiert und Substantive werden in Abhängigkeit des Falls dekliniert. Besitzt man eine statische Liste von möglichen MWA-Kandidaten, die mit dem Eingabetext verglichen wird, so muss sichergestellt werden, dass auch alle möglichen morphologischen Inflektionen des Ausdrucks in den Vergleich einbezogen werden. Wird dies nicht beachtet, so werden einige MWAs aufgrund grammatischer Veränderungen nicht erkannt, obwohl die Grundform des Ausdrucks eigentlich in der Liste vorhanden ist. Es ist zu umständlich alle möglichen Formen eines MWAs in der Liste zu führen. Eine mögliche Lösung ist es also lediglich die Stammformen der MWAs in der Liste aufzuführen. Jede Wortsequenz im Text, die mit der Liste verglichen werden soll, muss dann lediglich lemmatisiert werden.

Die Arbeit „UFRGS&LIF at SemEval-2016 Task 10: Rule-Based MWE Identification and Predominant-Supersense Tagging“ [CRV16] verwendet einen regelbasierten Ansatz, der sich von den bisherigen listenbasierten Ansätzen unterscheidet. Zunächst werden durch drei überwachte Regeln eine Menge von MWAs aus einem bereits annotierten Datensatz extrahiert und als Liste für die Identifikation verwendet. Anschließend wird durch drei Muster, die auf den Wortartmarkierungen von Wörtern basieren, eine Menge von MWA-Kandidaten aus dem Eingabetext extrahiert und zu der zuvor erstellten Liste hinzugefügt. Anschließend wird die erstellte Liste für die listenbasierte MWA-Identifikation verwendet. Im Gegensatz zu den anderen listenbasierten Ansätzen, verwendet das System keine Liste von MWAs aus einer Wissensquelle, sondern erstellt diese selber, anhand von Mustererkennung bei Wortartmarkierungen und der Untersuchung eines bereits annotierten Datensatzes. Dieser Ansatz erzielt mit einem F_1 -Wert von 56.75% [SHJC16] bei den TED talks die besten Ergebnisse der regelbasierten Ansätze bei der SemEval-2016. Es ist hierbei jedoch wichtig zu beachten, dass dieser regelbasierte Ansatz einen Korpus mit bereits annotierten MWAs verwendet, um einen Teil der Liste mit MWA-Kandidaten zu füllen. Da im Kontext dieser Arbeit vor allem mit Anforderungsspezifikationen gearbeitet wird und für diese kein großer Korpus mit annotierten MWAs existiert, ist eine Replikation der guten Ergebnisse nicht sicher. Das System müsste die MWA-Kandidaten lediglich aus den unmarkierten

Anforderungen mithilfe von Mustererkennung bei Wortartmarkierungen extrahieren. Eine Nutzung des Ansatzes kommt aufgrund der guten Ergebnisse dennoch in Betracht.

5.4.2 MWA-Identifikation durch maschinelles Lernen

Neben den regelbasierten Ansätzen für die Identifikation von MWAs existieren noch Verfahren, die auf dem Prinzip des maschinellen Lernens basieren. Hierfür werden in Textkorpora zunächst manuell die verschiedenen MWAs annotiert und anschließend als Trainingsdaten für das System verwendet. Im besten Fall werden für das Lernverfahren Textkorpora verwendet, in denen besonders viele MWAs vorkommen, da somit eine große Menge an MWAs erlernt werden kann. Im Gegensatz zu den regelbasierten Ansätzen, ist ein großer manueller Aufwand erforderlich, um eine gute Performanz zu erreichen. Außerdem muss im Kontext dieser Arbeit beachtet werden, dass die Trainingsdaten, vor allem aus Anforderungen bestehen müssen, damit ein präzises Training möglich ist und somit auch domänenspezifische Fachbegriffe identifiziert werden können.

In der Arbeit „UW-CSE at SemEval-2016 Task 10: Detecting Multiword Expressions and Supersenses using Double-Chained Conditional Random Fields“ [HSL16] wird ein doppelt verkettetes bedingtes Zufallsfeld für die MWA-Identifikation trainiert. Das doppelt verkettete Zufallsfeld wird jedoch nur verwendet, da die Aufgabe bei der SemEval-2016 zusätzlich zur MWA-Identifikation noch die Annotation von semantischen Kategorien fordert. Für die einfache MWA-Identifikation würde also ein einfach verkettetes Zufallsfeld ausreichen. Das verkettete Zufallsfeld wird anschließend mit einem passenden, manuell annotierten Trainingsdatensatz trainiert und kann dann für die Identifikation von MWAs verwendet werden. Das System erzielte die besten F_1 -Werte mit 54.80% auf Reviewtexten und 61.09% auf Tweets bei der SemEval-2016 [SHJC16]. Lediglich auf TED-talks konnten die Autoren Cordeiro, Ramisch und Villavicencio [CRV16] mit ihrem regelbasierten Ansatz einen besseren Wert erzielen.

Vergleicht man nun die Ergebnisse der beschriebenen verwandten Arbeiten, so wird deutlich, dass die MWA-Identifikation durch maschinelles Lernen die besten Ergebnisse erzielt. Das System UTU [BS16] schneidet mit dem listenbasierten Ansatz und einem maximal erreichten F_1 -Wert von 20.11% deutlich schlechter ab als alle anderen Ansätze und wird deshalb nicht verwendet. Trotz des hohen F_1 -Wertes den das System der Autoren Arranz, Atserias und Castillo [AAC05] erzielt, liegt die Vermutung nahe, dass diese aufgrund des sehr spezifischen Evaluationsdatensatzes, nicht bei Anforderungsspezifikationen replizierbar sind. Eine Verwendung kommt im Kontext dieser Arbeit also nicht infrage. Obwohl das System UW-CSE [HSL16] bei den Reviews und Tweets mit den F_1 -Werten 53.80% und 61.09% besser ist als alle anderen Systeme, schneidet das System UFRGS&LIF [CRV16] bei den TED talks mit einem F_1 -Wert von 56.76% besser ab. Sowohl das System UW-CSE, als auch das System UFRGS&LIF arbeiten bei ihrem Ansatz mit einem bereits annotierten Textkorpus. Da für die Anforderungsspezifikationen ein solcher Korpus nicht existiert, muss dieser zunächst erstellt werden, weshalb der manuelle Aufwand beider Systeme gleich ist. Aufgrund der besseren Ergebnisse in zwei von drei Bereichen und des äquivalenten manuellen Aufwands, wird in dieser Arbeit deshalb ein Ansatz, der sich an dem System UW-CSE orientiert, für die MWA-Identifikation verwendet.

5.5 Wortbedeutungsauflösung

Die Bedeutungsauflösung von Wörtern unter Berücksichtigung von MWAs ist das dritte Teilziel dieser Arbeit und wird im folgenden Abschnitt näher betrachtet. Ein WSD-System dieser Art soll bei einzelnen Wörtern nicht anders verfahren als andere klassische WSD-Systeme. Liegt jedoch ein Eingabetext vor in dem MWAs markiert sind, so soll das WSD-System versuchen eine Bedeutung für den gesamten MWA zu finden.

Beispiel 5.5: Beispielsatz bei dem die MWAs bereits markiert wurden

„[The] [user] [can] [access] [the] [system properties].“

Beispiel 5.5 zeigt einen Eingabetext bei dem sowohl isolierte Wörter, als auch der MWA „system properties“ markiert sind. Die Methode zur Bedeutungsauflösung soll versuchen dem gesamten Ausdruck „system properties“ eine passende Bedeutung zuzuordnen. Ist dies nicht möglich, soll das Verfahren die einzelnen Teilworte des Ausdrucks untersuchen, um mögliche Rückschlüsse auf die Bedeutung des gesamten Ausdrucks zu ziehen.

5.5.1 Wissensbasierte Wortbedeutungsauflösung

Wie in Kapitel 2 bereits genauer beschrieben, arbeiten wissensbasierte WSD-Systeme mit den Informationen aus bereits vorhandenen, manuell erstellten Wissensquellen, um eine Wortbedeutungsauflösung durchzuführen. Hierfür werden verschiedene Definitionen des Zielwortes aus den Wissensquellen extrahiert und anschließend wird anhand des Kontexts, in dem das Wort auftritt, eine passende Bedeutung ausgewählt. Ähnlich wie bereits bei der regelbasierten MWA-Erkennung in Abschnitt 5.4.1, ist der Erfolg des Verfahrens eng mit der Anzahl an Informationen, die in der Wissensquelle vorhanden sind, verknüpft. Da das zu entwickelnde Verfahren dieser Arbeit vor allem auf Anforderungsspezifikationen arbeiten soll, treten Situationen auf in denen auch domänenspezifischen Fachbegriffen eine Bedeutung zugeordnet werden muss. Für eine erfolgreiche wissensbasierte Bedeutungsauflösung muss also eine Wissensquelle gewählt werden, die auch Definitionen solcher Fachbegriffe enthält. Bei der Nutzung öffentlicher Wissensquellen wie WordNet besteht beispielsweise bei Fachbegriffen die Gefahr, dass eventuell keine Definition, oder zumindest nicht die korrekte Definition für den gegebenen Kontext vorhanden ist.

Alle drei verwandten Arbeiten, die auf der wissensbasierten Wortbedeutungsauflösung basieren [AAC05] [ALS18] [CS18], beschreiben Systeme, die mit der öffentlichen Wissensquelle WordNet arbeiten. Durch diese Einschränkung entstehen möglicherweise Informationslücken, die sich negativ auf das bereits angesprochene Problem der domänenspezifischen Fachbegriffe, auswirken. Ein möglicher Ansatz, wäre stattdessen die Nutzung von Wikipedia, da diese aufgrund der Bearbeitung durch viele Personen, stetig wächst und somit eventuell auch mehr fachspezifische Informationen enthält. Wichtig ist hierbei zu beachten, dass Wikipedia vor allem Nomen und Eigennamen als Einträge enthält, weshalb eine Disambiguierung von anderen Wortarten wie Verben nicht möglich ist.

Das Grundprinzip der wissensbasierten Ansätze ist in vielen Fällen ähnlich. Bei einem gegebenen Eingabekontext wird versucht eine Menge von Wörtern zu disambiguieren. Jedes der zu disambiguierenden Wörter enthält eine Menge von möglichen Bedeutungen in der verwendeten Wissensquelle. Bei der Bedeutungsauflösung wird dann der Kontext, in dem das Wort auftritt, verwendet, um eine der möglichen Bedeutungen aus der Wissensquelle auszuwählen. Die Verfahren unterscheiden sich meist in der Größe des verwendeten Kontextes und der Art und Weise, wie der Kontext für die Auswahl einer Bedeutung verwendet wird.

In der Arbeit „Multiwords and Word Sense Disambiguation“ [AAC05] wird ein System zur wissensbasierten Wortbedeutungsauflösung mit WordNet beschrieben. Für die Auswahl einer passenden Bedeutung für ein Wort, werden 11 Heuristiken verwendet, die auf den bereits vorhandenen Funktionen der Wissensquelle WordNet basieren. Jede der Heuristiken gibt dabei ein oder mehrere Stimmen für die verschiedenen möglichen Bedeutungen des Wortes ab. Die Bedeutung mit den meisten Stimmen wird schlussendlich dem Wort

zugeordnet. Beispielsweise gibt die „Most Frequent“ Heuristik jeder Bedeutung eines Wortes, deren Häufigkeit mindestens 85% der Häufigkeit der häufigsten Bedeutung entspricht, eine Stimme ab. Die häufigste Bedeutung wird in dieser Heuristik mit einbezogen. Schlussendlich wird die Bedeutung, die durch die 11 Heuristiken die meisten Stimmen erhalten hat, ausgewählt. Ein großer Nachteil dieses Verfahrens ist, dass es für die Bedeutungsauflösung von WordNet Glossen konzipiert wurde. Dem System ist deshalb bekannt zu welchem Synset die zu disambiguierende Glosse zugeordnet ist. Viele der Heuristiken des Systems basieren auf dieser Information, weshalb eine direkte Übertragung auf einen normalen Eingabetext nicht möglich ist. Aufgrund der spezifischen Anpassung des Systems auf die Disambiguierung von WordNet Glossen, kommt eine Nutzung des Ansatzes für die Bedeutungsauflösung bei Anforderungen nicht infrage.

In der Arbeit „The risk of sub-optimal use of Open Source NLP Software: UKB is inadvertently state-of-the-art in knowledge based WSD“ [ALS18] beschreiben die Autoren, einen graphbasierten Ansatz mit der Wissensquelle WordNet. Bei der Bedeutungsauflösung verwendet ihr System den Graphen, der WordNet zugrunde liegt. Dabei wird jedes zu disambiguierende Wort des Eingabekontextes mit seinen entsprechenden Konzepten im Graph verbunden. Anschließend wird ein personalisierter PageRank berechnet. Das Konzept mit dem maximalen Rang wird dem Wort anschließend als Bedeutung zugeordnet. Mit den richtigen Parametern erreicht das System bei der SemEval-2015 einen F_1 -Wert von 70.3 und schneidet somit, abgesehen von Babelfy, besser ab als andere wissensbasierte Systeme. Bei der Wahl der Parameter spielt beispielsweise die Größe des Kontextfensters eine Rolle. Insgesamt schlagen die Autoren vor bei der Nutzung ihres WSD-Systems UKB mindestens 20 Worte, als Kontext für die Wortbedeutungsauflösung in Betracht zu ziehen. Wichtig ist zu erwähnen, dass das UKB-System sich nicht auf die Wissensquelle WordNet beschränkt, sondern erlaubt theoretisch jeden möglichen Graph zu verwenden. Dabei muss jedoch beachtet werden, dass nicht jeder Graph sich für die Wortbedeutungsauflösung eignet. Für die Analyse ist deshalb relevant zu untersuchen, ob die Graphenstruktur anderer Wissensquellen, wie beispielsweise Wikipedia, für die Bedeutungsauflösung mit UKB geeignet sind. Falls dies der Fall ist, so lassen sich durch Nutzung von UKB die Einschränkungen der Wissensquelle WordNet, also fehlende Definitionen für Fachbegriffe und MWAs, umgehen.

Die Arbeit „Knowledge-Based Word Sense Disambiguation Using Topic Models“ [CS18] geht noch einen Schritt weiter und versucht das gesamte Dokument als Kontext für die Wortbedeutungsauflösung zu betrachten. Hierfür wird ein komplexes Themenmodell verwendet, um die Dokumente anhand von Synset Wahrscheinlichkeiten zu repräsentieren. Bei der Evaluation wird jedoch deutlich, dass dieser Ansatz beispielsweise mit einem F_1 -Wert von 69.6 bei der SemEval-2015 nicht besser abschneidet, als das zuvor erwähnte UKB-System, welches deutlich weniger Kontextinformationen verwendet.

Betrachtet man die wissensbasierten Ansätze allgemein für die Bedeutungsauflösung mit Berücksichtigung der MWAs, so wird deutlich, dass sich das Vorgehen nur gering von der Wortbedeutungsauflösung einzelner Wörter unterscheidet. Falls eine Definition für einen markierten MWA in der verwendeten Wissensquelle vorhanden ist, so ändert sich die Vorgehensweise im Vergleich zu einem einzelnen Wort nicht. Der MWA wird dem System übergeben und es wird unter den vorhandenen Definitionen, anhand des bereitgestellten Kontextes eine passende Bedeutung ausgewählt. Lediglich für die MWAs bei denen keine Definition in der Wissensquelle vorhanden ist, ändert sich die Vorgehensweise. In diesem Fall muss untersucht werden, ob auch für Teile des MWAs eine Bedeutung vorhanden ist, die Rückschlüsse auf die Bedeutung des gesamten MWAs zulässt.

Insgesamt stellt die wissensbasierte Wortbedeutungsauflösung vor allem aufgrund des geringen manuellen Aufwands eine gute Lösungsmöglichkeit für diese Arbeit dar. Die Per-

formanz dieser Systeme bei MWAs ist stark abhängig davon, für wie viele MWAs eine passende Definition in der verwendeten Wissensquelle vorhanden ist. Das Vorgehen der Bedeutungsauflösung bei MWAs mit wissensbasierten Ansätzen, ist jedoch einfach umzusetzen, da es sich kaum von dem Vorgehen bei einzelnen Wörtern unterscheidet. Des Weiteren muss beachtet werden, dass entweder eine Wissensquelle mit ausreichenden domänenspezifischen Informationen ausgewählt wird, oder eine Kombination von Wissensquellen verwendet wird, die eine ausreichende Menge an Informationen bietet.

5.5.2 Überwachte Wortbedeutungsauflösung

Der Ansatz der überwachten Wortbedeutungsauflösung schneidet, wie bereits in Kapitel 4 festgestellt wurde, bei Evaluationen in fast allen Fällen besser ab als die wissensbasierten Ansätze. Das Grundprinzip hierbei ist, dass Techniken des maschinellen Lernens verwendet werden, um mithilfe von manuell annotierten Daten verschiedene Wortbedeutungen und den zugehörigen Kontext zu erlernen. Die Performanz eines Systems, das maschinelles Lernen verwendet, ist stark abhängig von der Menge der Daten, die für den Lernprozess bereitgestellt werden. Da überwachte Lerntechniken mit manuell annotierten Daten arbeiten, sind diese immer mit einem hohen manuellen Arbeitsaufwand verbunden. Für viele allgemeine überwachte WSD-Systeme existieren jedoch bereits annotierte Textkorpora, wie beispielsweise das SemCor-Korpus für die englische Sprache, die diesen Vorgang vereinfachen. Vor allem bei dieser Arbeit gilt es jedoch zu beachten, dass das zu entwickelnde WSD-System auf MWAs in Anforderungsdokumenten spezialisiert sein soll. Für ein präzises Training sind also Trainingsdaten nötig, die aus Anforderungen erstellt wurden und eine große Menge an MWAs und domänenspezifischen Fachbegriffen enthalten. Da solch ein Datensatz nicht frei verfügbar ist, muss dieser zunächst manuell erstellt werden. Außerdem ist wichtig zu beachten, dass Sprache sich im Laufe der Zeit entwickelt und somit immer wieder neue Begriffe auftauchen, für die eine Bedeutungsauflösung durchgeführt werden muss. Die wissensbasierten Ansätze haben somit den Vorteil, dass lediglich die Wissensbasis erweitert, oder ausgetauscht werden muss, um auch bei neuen Begriffen eine passende Bedeutungsauflösung durchführen zu können. Die überwachten Ansätze hingegen benötigten weitere Trainingsdaten, um die passenden Definitionen für die neuen Begriffe zu erlernen. Es stellt sich also die Frage, ob die überwachten Lernverfahren mit der beschränkten Anzahl an Trainingsdaten eine Leistungssteigerung erzielen können, die den benötigten Mehraufwand rechtfertigen.

Die Arbeit „SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation“ [PRDB17] stellt ein flexibles und schnelles Java Toolkit vor, welches für die überwachte Wortbedeutungsauflösung verwendet wird. Es werden verschiedene Textverarbeitungsmodule bereitgestellt, die eine Vorverarbeitung von Textkorpora durchführen, um anschließend einen Klassifikator zu trainieren. Vor allem wegen der Geschwindigkeit und des modularen Aufbaus des Systems bietet es eine gute Umgebung für überwachte WSD-Methoden. Wie bereits oben erwähnt, erzielen überwachte Lernverfahren nur gute Ergebnisse, wenn eine ausreichend große Menge an Trainingsdaten verwendet wird. Da SupWSD für das Training des Klassifikators auf bereits vorhandenen annotierte Textkorpora angewiesen ist, muss für die Anwendung des Systems eine Menge von Trainingsdaten bereitgestellt werden. Aufgrund der spezifischen Aufgabe dieser Arbeit gibt es jedoch keine vorgefertigten Textkorpora, weshalb dieser zunächst manuell erstellt werden müsste. Bei der SemEval-2015 erreichte das System einen maximalen F_1 -Wert von 71.6% und schneidet somit nicht viel besser ab als das wissensbasierte UKB-System.

In der Arbeit „GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge“ [HSQH19] wird ein Ansatz der überwachten Wortbedeutungsauflösung vorgestellt, der mit der öffentlichen Wissensquelle WordNet arbeitet. Die Autoren merken an, dass traditionelle überwachte Lernverfahren öffentliche Wissensquellen wie WordNet ignorieren und

somit zwar bessere Ergebnisse als die wissensbasierten Ansätze erzielen, aber damit auch in manchen Fällen weniger flexibel sind. Durch die Nutzung von WordNet und einem bereits vortrainierten BERT-Modell, sollen die Nachteile von überwachten WSD-Verfahren verbessert werden. Das System liefert mit einem maximalen F_1 -Wert von 80.4% bei der SemEval-2015 die besten Ergebnisse. Es ist jedoch wichtig zu beachten, dass BERT, obwohl es bereits vortrainiert ist, dennoch mit weiteren Trainingsdaten für die spezifische Aufgabe feinjustiert werden muss. In ihrer Arbeit verwenden die Autoren hierfür das SemCor3.0 Trainingskorpus. Für die spezielle Aufgabe dieser Arbeit müsste also auch hier ein Trainingsdatensatz aus Anforderungen erstellt werden.

Betrachtet man überwachte Ansätze allgemein für die Bedeutungsauflösung mit Berücksichtigung von MWAs so fallen einige Unterschiede zu den wissensbasierten Ansätzen auf. Ein überwachter Ansatz lernt die Bedeutungen von Ausdrücken anhand der Annotationen innerhalb eines Trainingsdatensatzes. Falls in diesem Trainingsdatensatz ein MWA enthalten ist, dann lernt das System die entsprechende Bedeutung und kann diese in anderen Texten weiterverwenden. Es tritt jedoch ein Problem auf, falls ein neuer MWA in dem Eingabetext erscheint und das System keine entsprechende Bedeutung aus dem Datensatz erlernt hat. In diesem Fall sollte das System versuchen anhand der Teilbedeutungen Rückschlüsse auf die Bedeutung des gesamten MWAs zu ziehen. Dies ist jedoch nur möglich, wenn das System auch die Teilbedeutungen anhand des Trainingsdatensatzes erlernt hat. Dies kann allerdings nicht immer garantiert werden, weshalb die Vermutung gerechtfertigt ist, dass ein überwachter Ansatz, vor allem bei neuen MWAs häufig keine Bedeutung zuordnet. Ein wissensbasiertes System hingegen kann in diesem Fall einfach in der Wissensbasis nach Bedeutungen für Teile des MWAs suchen.

Insgesamt lässt sich also feststellen, dass für die effiziente Nutzung von traditionellen überwachten WSD-Methoden zunächst eine ausreichend große Menge an Trainingsdaten vorhanden sein müssen. Da die spezifische Aufgabe dieser Arbeit, die mögliche Menge an Trainingsdaten stark einschränkt, sind Toolkits wie SupWSD weniger geeignet. Obwohl GlossBERT bereits mit einer großen Menge an Daten vortrainiert ist und sehr gute Ergebnisse liefert, muss für die Feinjustierung dennoch ein weiterer Trainingsdatensatz bereitgestellt werden. Es ist schwierig einzuschätzen, ob auch ein kleiner annotierter Datensatz ausreichen würde, um gute Ergebnisse mit Hilfe von GlossBERT zu erzielen. Trotz der guten Ergebnisse bei klassischen WSD-Aufgaben, scheinen die überwachten Ansätze, vor allem wegen der fehlenden Trainingsdaten und der geringen Flexibilität im Bezug auf die Bedeutungsauflösung bei MWAs im Kontext dieser Arbeit eine schlechtere Wahl zu sein, als die wissensbasierten Ansätze.

5.5.3 Unüberwachte Wortbedeutungsauflösung

Wie bereits in Kapitel 2 näher beschrieben, heben sich die Ansätze der unüberwachten Wortbedeutungsauflösung deutlich von den überwachten und wissensbasierten Ansätzen ab. Sowohl überwachte, als auch wissensbasierte WSD-Methoden ordnen jedem Wort eine fest definierte Bedeutung zu, wodurch ein Vergleich der Effektivität beider Ansätze möglich wird. Unüberwachte WSD-Methoden hingegen lernen Informationen aus nicht annotierten Datensätzen, bei denen Wortbedeutungen zuvor nicht manuell festgelegt wurden. Sie haben somit zunächst die Aufgabe, die möglichen Bedeutungen eines Wortes mit Hilfe der Daten zu induzieren. Die Wortbedeutungen, die auf diese Weise entstehen, werden meist in Form von Wortclustern repräsentiert und sind somit deutlich schwieriger zu interpretieren, als die festen Definitionen, die die überwachten und wissensbasierten Ansätze verwenden. Wie bereits in Abschnitt 5.5.2 erwähnt, gibt es aufgrund der spezifischen Aufgabe dieser Arbeit einen Mangel an annotierten Trainingsdaten, die für die überwachte Wortbedeutungsauflösung nötig wären. Dieses Problem ist bei einem unüberwachten Ansatz nicht gegeben, da Anforderungsdokumente ohne Annotationen für das Training verwendet werden können.

Hierbei ist es jedoch wichtig zu beachten, dass die Ergebnisse, die solch ein System erzielen würde, aufgrund der geringen Interpretierbarkeit keine angemessene Lösung im Rahmen des INDIRECT-Projektes darstellen.

Beispiel 5.6: Beispiel für eine Wortbedeutung bei unüberwachten Systemen [PABP17]

table (furniture)

- counter, console, bench, dinner, table, dining table, desk, surface, bar, board

table (data)

- chart, list, index, graph, columned list, tabulation, standings, diagram, ranking

table (geo)

- level, plateau, plain, flatland, saturation level, water table, geographical level, water level

Beispiel 5.6 zeigt mehrere Wortcluster des Begriffs „table“ aus der Arbeit „Making Sense of Word Embeddings“ [PABP17]. Das in der Arbeit beschriebene System führt zunächst eine Wortbedeutungsinduktion (engl. word sense induction (WSI)) durch, um aus nicht annotierten Daten die im Beispiel gezeigten Cluster zu erzeugen. Diese Cluster sind eine Menge von Synonymen und werden von dem System als Bedeutungsrepräsentationen verwendet. Wie deutlich erkennbar ist, sind die Lösungen die das System erstellt nur schlecht interpretierbar im Vergleich zu wissensbasierten und überwachten Ansätzen. Ein wissensbasierter Ansatz, der mit der Wissensquelle WordNet arbeitet, hat beispielsweise Zugriff auf Daten wie: genaue Definitionen der Begriffe, Häufigkeiten, oder auch Relationen zu anderen Begriffen und Wortarten [Mil95]. Wie bereits in Kapitel 3 genauer erklärt, erstellt das Projekt INDIRECT ein Absichtenmodell in Form eines Graphen aus den Anforderungsspezifikationen. Dieser Graph soll die semantischen Beziehungen zwischen den Elementen der Anforderungen darstellen. Mithilfe der genannten Informationen aus der Wissensquelle WordNet lassen sich beispielsweise Über- und Unterkonzepte, Beziehungen zwischen den Elementen oder auch Ähnlichkeiten einfacher erkennen. Ein unüberwachter Ansatz, der Lösungen erzeugt, aus denen INDIRECT nur wenig Informationen gewinnen kann, eignet sich deshalb im Kontext dieser Arbeit nicht und wird deshalb auch nicht weiter ausgeführt.

5.6 Wissensbasierte Bedeutungsauflösung bei MWAs

Durch die Ergebnisse der Analyse lässt sich also feststellen, dass lediglich wissensbasierte und überwachte Bedeutungsauflösungsansätze für das zu entwickelnde WSD-System infrage kommen. Da nicht abschätzbar ist, ob eine geringe Menge von Trainingsdaten für das überwachte GlossBERT System ausreicht, um gute Ergebnisse zu erzielen, und für andere konventionelle überwachte Ansätze wie beispielsweise SupWSD nicht genügend annotierte Trainingsdaten vorhanden sind, scheint ein wissensbasierter Ansatz die beste Lösung im Kontext dieser Arbeit darzustellen. Für einen wissensbasierten Ansatz muss zunächst eine Wissensquelle ausgewählt werden, deren Wortbedeutungen später auf die Token des Eingabetextes abgebildet werden. WordNet ist eine weit verbreitete Wissensquelle, die häufig im Kontext der Wortbedeutungsauflösung verwendet wird und eine Vielzahl an semantischen Relationen, wie Hyponymie, Hypernomie, syntaktische Kategorien, etc. bereitstellt [Mil95]. Da das zu entwickelnde System jedoch vor allem auf Anforderungen arbeiten soll, muss häufig eine Bedeutungsauflösung bei Fachbegriffen, oder auch bei MWA-Fachbegriffen durchgeführt werden. Es besteht bei der Wahl von WordNet also das Risiko, dass viele der benötigten Wortbedeutungen nicht in der Wissensbasis vorhanden

sind. Im Kontext dieser Arbeit stellt Wikipedia, aufgrund ihres stetigen Wachstums und der Bearbeitungsmöglichkeit durch eine große Bandbreite an Personen, vermutlich eine bessere Wahl dar.

5.6.1 Wikipedia für die wissensbasierte Bedeutungsauflösung

Bei der Wahl von Wikipedia als Wissensquelle für die Bedeutungsauflösung muss zunächst untersucht werden, inwiefern diese sich für die Aufgabe eignet, und welche wissensbasierten Systeme mit Wikipedia arbeiten können. Das bereits beschriebene UKB-System [AS09] beschränkt sich beispielsweise nicht auf eine einzige Wissensquelle, sondern arbeitet generell mit Wissensbasen, die eine zugrunde liegende Graphstruktur besitzen. Da das System neben Babelfy die besten Ergebnisse im Bereich der wissensbasierten Bedeutungsauflösung erzielt, wird es in dieser Arbeit für die Bedeutungsauflösung in Betracht gezogen. Um zu erfahren, ob eine Kombination von Wikipedia und UKB möglich ist, muss zunächst untersucht werden, ob Wikipedia eine Graphstruktur besitzt, die ähnlich wie der WordNet Graph, relevante Ergebnisse im Bereich der Bedeutungsauflösung erzielen kann. Hierfür wird zunächst betrachtet, welche Relationen durch den WordNet Graphen [Mil95] dargestellt werden, und ob ähnliche Strukturen innerhalb von Wikipedia gefunden werden können. Die Hauptkonzepte und somit die Knoten des WordNet Graphen werden durch die sogenannten Synsets gebildet. Die Synsets sind Mengen von Wörtern mit gleicher Bedeutung und repräsentieren somit die Synonymie-Beziehung zwischen den Wörtern der Menge. Zwischen diesen Synsets existieren zusätzlich weitere Beziehungen, wie die Hypernymie, Antonymie, Meronymie und Troponymie, die die Kanten des WordNet Graphen bilden. Um Zugriff auf die Daten von Wikipedia in strukturierter Form zu bekommen, bietet sich die Wissensbasis DBpedia an, die bereits in Kapitel 2 näher erläutert wurde. DBpedia besitzt ein Extraktionsframework, dass die Wikipedia Seiten nach Informationen durchsucht und diese innerhalb von Datensätzen in Form von RDF Aussagen speichert [LIJ⁺15]. Betrachtet man beispielsweise die Beziehung „Hypernymie“, so ist diese eine Relation zwischen Unter- und Oberbegriffen. Möchte man ähnliche Relationen in DBpedia finden, so sollte deshalb nach Datensätzen gesucht werden, die Unterkonzepte zu ihren Oberkonzepten zuordnen. Ein wichtiger Datensatz, der in diesem Zusammenhang existiert ist das „Linked Hypernym Dataset (LHD)“ [Kli15], welcher DBpedia Einträge zu ihren Hypernymen zuordnet.

Beispiel 5.7: Beispiel für RDF Aussagen aus dem Linked Hypernym Dataset [DBp]

Triple 1:

```
< http://dbpedia.org/resource/Tebra_castle >  
< http://www.w3.org/1999/02/22-rdf-syntax-ns#type >  
< http://dbpedia.org/ontology/Castle > .
```

Triple 2:

```
< http://dbpedia.org/resource/Coal_breaker >  
< http://purl.org/linguistics/gold/hypernym >  
< http://dbpedia.org/resource/Plant > .
```

Beispiel 5.7 zeigt die zwei verschiedenen Arten von RDF Aussagen aus dem „Linked Hypernym Dataset“. Die erste Art von RDF Aussage ordnet einem DBpedia Eintrag jeweils ein Hypernym aus der DBpedia Ontologie zu. Die zweite Art von RDF Aussage ordnet

einem DBpedia Eintrag jeweils einen anderen DBpedia Eintrag als Hypernym zu. Beide Relationen entsprechen einer Zuordnung von Unterkonzepten zu Oberkonzepten und sollten deshalb bei der Bedeutungsauflösung in gleicher Weise wie die Hypernymie-Relation des WordNet Graphen agieren. Des Weiteren existiert der sogenannte „Categories“ Datensatz [ABK⁺07], der eine Hierarchie aus den DBpedia Einträgen und den zugehörigen Kategorien aufbaut.

Beispiel 5.8: Beispiel für RDF Aussagen aus dem Categories Datensatz [DBp]

Triple 1:

```
< http://dbpedia.org/resource/An_American_in_Paris >
< http://purl.org/dc/terms/subject >
< http://dbpedia.org/resource/Category:1928_compositions > .
```

Triple 2:

```
< http://dbpedia.org/resource/Category:Middle-earth_Dwarves >
< http://www.w3.org/2004/02/skos/core#broader >
< http://dbpedia.org/resource/Category:Middle-earth_characters > .
```

Triple 3:

```
< http://dbpedia.org/resource/Category:Software >
< http://www.w3.org/2004/02/skos/core#related >
< http://dbpedia.org/resource/Category:Application_software > .
```

Beispiel 5.8 zeigt drei verschiedene Arten von RDF Aussagen aus dem Categories Datensatz. Die erste Art von Aussage enthält das Prädikat „`< http://purl.org/dc/terms/subject >`“ [dcm], welches verwendet wird, um darzustellen, dass ein DBpedia Eintrag, das Thema einer Kategorie ist. Dies ist vergleichbar mit einer Zuordnung von Unterkonzepten zu Oberkonzepten und somit auch vergleichbar mit der Hypernymhierarchie des WordNet Graphen. Die zweite Art von Aussage verwendet das Prädikat „`< http://www.w3.org/2004/02/skos/core#broader >`“, welches eine Kategorie zu einer allgemeineren Kategorie zuordnet [MMWB05]. Durch diese Zuordnung entsteht eine Hierarchie, die mit der Hypernymhierarchie des WordNet Graphen ebenfalls vergleichbar ist. Die letzte Art von Aussage verwendet das Prädikat „`< http://www.w3.org/2004/02/skos/core#related >`“, welches eine assoziative Relation zwischen zwei Kategorien darstellt [MMWB05]. Diese Relation ist nicht direkt abbildbar auf eine der Relationen des WordNet Graphen. Da sie jedoch ebenfalls eine semantische Beziehung zwischen den Kategorien darstellt, sollte eine Nutzung im Zusammenhang mit dem UKB-System ebenfalls möglich sein.

Insgesamt ist feststellbar, dass die Wissensbasis DBpedia eine große Menge an Relationen und Konzepten besitzt, deren Struktur zumindest die Hypernymhierarchie des ursprünglich verwendeten WordNet Graphen widerspiegelt. Obwohl die anderen Relationen des WordNet Graphen nicht direkt abbildbar auf den DBpedia Graphen zu sein scheinen, sollte dieser dennoch relevante Ergebnisse bei der Bedeutungsauflösung erzeugen können. Diese Vermutung basiert vor allem darauf, dass die Hypernymierelation, die häufigste und somit relevanteste Relation des WordNet Graphen darstellt [wora].

5.6.2 Bedeutungsauflösung mit Berücksichtigung von MWAs

Nachdem nun festgestellt wurde, dass eine Kombination des UKB-Systems mit der Wissensquelle DBpedia möglich sein sollte, muss das allgemeine Vorgehen bei der Bedeutungsauflösung mit Berücksichtigung von MWAs näher betrachtet werden. Wie bereits zu Beginn von Abschnitt 5.5 beschrieben wurde, soll die Bedeutungsauflösung mit Eingabetexten arbeiten, in denen die vorhandenen MWAs bereits markiert sind. Da die Analyse des vorherigen Abschnitts bereits ergeben hat, dass Wikipedia als Wissensquelle für die Bedeutungsauflösung verwendet werden soll, muss beachtet werden, dass Wikipedia hauptsächlich Einträge zu Nomen und Eigennamen besitzt. Im Kontext dieser Arbeit stellt diese Einschränkung jedoch nur ein geringes Problem dar, da die für INDIRECT relevanten Informationen vor allem in den Nomen und Eigennamen enthalten sind. Eine Bedeutungsauflösung der anderen Wortarten ist deshalb nicht zwingend notwendig. Um dennoch die Lücke an Informationen zu füllen, die durch die fehlende Bedeutungsauflösung bei den anderen Wortarten entsteht, kann die bereits mit UKB getestete Wissensquelle WordNet verwendet werden, um die restlichen Worte zu disambiguieren. Da die anderen Wortarten wenig Informationen bezüglich der Anforderungen tragen und somit auch nicht anforderungsspezifisch sind, sollte WordNet passende Ergebnisse erzielen.

Im nachfolgenden Abschnitt wird die Vorgehensweise der Bedeutungsauflösung bei einzelnen Wörtern und MWAs genauer beschrieben.

1. Fall: Aufzulösender Ausdruck besteht aus einem Wort

Dieser Fall unterscheidet sich nicht von der klassischen Vorgehensweise bei wissensbasierten WSD-Systemen. Der Ausdruck wird dem System übergeben und es wird nach vorhandenen Definitionen gesucht. Ist keine Definition vorhanden, so bekommt der Ausdruck keine Bedeutung zugewiesen. Ist genau eine Bedeutung vorhanden, so wird diese dem Ausdruck zugewiesen. Falls mehrere mögliche Bedeutungen gefunden werden, so muss der Kontext der entsprechenden Anforderung, in dem das Wort vorkommt, für die Auswahl einer möglichen Definition in Betracht gezogen werden. Dies entspricht dem klassischen WSD-Verfahren bei einzelnen Wörtern.

2. Fall: Aufzulösender Ausdruck besteht aus einem MWA

Ist der zu disambiguierende Ausdruck ein MWA, so sollte zunächst versucht werden in der Wissensquelle eine Bedeutung für den gesamten Ausdruck zu finden, da nur diese die vollständigen Informationen über den MWA enthält. Wie bereits zuvor beschrieben beinhaltet, der in dieser Arbeit verwendete Begriff „MWA“, lediglich die zusammengesetzten Nomen und Eigennamen. Da, wie bereits in Absatz 5.2.1 beschrieben, für die Eigennamen keine festen Regeln bezüglich ihrer Veränderbarkeit festgelegt wurden, sollte der Eigenname ohne Veränderung in der Wissensquelle gesucht werden. Die zusammengesetzten Nomen hingegen können in Plural - oder Singularform auftreten. Damit der Ausdruck in Singularform im System gesucht wird, sollte dieser zunächst lemmatisiert werden. Falls mindestens eine Bedeutung für den Ausdruck gefunden werden kann, so kann der MWA wie ein einzelnes Wort behandelt werden und das weitere Vorgehen ist identisch zu Fall 1.

Enthält die Wissensquelle jedoch keine Definition für den gesamten MWA, so können generell zwei Vorgehensweisen unterschieden werden. Zum einen besteht die Möglichkeit zu entscheiden, dass dem Ausdruck aufgrund der fehlenden Definition in der Wissensquelle keine Bedeutung zugewiesen wird. Dies sorgt dafür, dass die MWAs nur dann eine Bedeutung zugewiesen bekommen, wenn diese auch vollständig ist. Hierbei entsteht jedoch der Nachteil, dass gewisse Informationen verloren gehen, die von dem INDIRECT-System

genutzt werden könnten. Angenommen das zusammengesetzte Nomen „system administrator“ besitzt keine vollständige Bedeutung in der Wissensquelle. Für das Teilwort „administrator“ hingegen ist eine passende Bedeutung vorhanden. Die Bedeutung des Wortes „administrator“ überschneidet sich zumindest teilweise mit der des Ausdrucks „system administrator“, da dieser lediglich eine spezifischere Form darstellt. Wenn also dem gesamten Ausdruck die Bedeutung des Wortes „administrator“ zugewiesen wird, dann wird zumindest deutlich, dass es sich um einen Administrator handelt und man verliert lediglich die Informationen, welche spezifische Aufgabe dieser Administrator besitzt. Im Gegensatz dazu gehen alle Informationen verloren, wenn nicht einmal eine Teilbedeutung zugewiesen wird. Die bessere Vorgehensweise ist also nach Teildefinitionen für die MWAs in der Wissensquelle zu suchen. Hierfür muss zunächst analysiert werden, welche Bestandteile eines MWAs für das Bestimmen einer Teilbedeutung geeignet sind. Die zusammengesetzten Nomen gehören zur Klasse der semi-festen Ausdrücke und stellen eine Sequenz von aufeinanderfolgenden Nomen dar. In der Arbeit „On the interpretation of noun compounds: Syntax, semantics, and entailment“ [Nak13] werden die Eigenschaften dieser Ausdrücke näher beschrieben. Der Autor unterscheidet hierbei zwischen endozentrischen und exozentrischen zusammengesetzten Nomen. Die endozentrischen Nomen bestehen aus einem Kopf und einem modifizierenden Teil, der die Eigenschaften des Kopfes verändert, während die exozentrischen Nomen keinen Kopf besitzen. Falls es sich bei dem Ausdruck um ein endozentrisches Nomen handelt, so ist wichtig, dass bei der Suche von Teilbedeutungen immer der Kopf in die Suche einbezogen wird, da dieser die eigentliche Bedeutung trägt und alle anderen Bestandteile lediglich dessen Bedeutung modifizieren. Handelt es sich bei dem Ausdruck, um ein exozentrisches Nomen, so kann nicht direkt bestimmt werden, welche Bestandteile relevant für die Bedeutung sind.

Die Eigennamen stellen eine weitere Klasse der semi-festen Ausdrücke dar, bei denen es schwierig ist, die eigentliche Bedeutung des Ausdrucks anhand der einzelnen Teilworte festzulegen.

Beispiel 5.9: Beispiel für einen Eigennamen mit passender Teilbedeutung

- Mozilla Firefox 3.0 - Teilbedeutung: Mozilla Firefox
- Microsoft Internet Explorer - Teilbedeutung: Internet Explorer

Beispiel 5.9 zeigt einige Beispiele für Eigennamen, bei denen die Suche nach Teilbedeutungen sinnvoll ist. Nimmt man an, dass die Wissensquelle keine Bedeutungen für die spezifischen Ausdrücke „Mozilla Firefox 3.0“ und „Microsoft Internet Explorer“ beinhaltet, dann sind auch die Teilbedeutungen zu den Ausdrücken „Mozilla Firefox“ und „Internet Explorer“ ausreichend, um die Bedeutung des gesamten Ausdrucks einzufangen.

5.7 Wortbedeutungsauflösung durch UKB

Das bereits beschriebene UKB-System bietet ein graphbasiertes Verfahren zur Wortbedeutungsauflösung bei Eingabetexten. In ihrer Arbeit „Personalizing PageRank for Word Sense Disambiguation“ [AS09] beschreiben die Autoren verschiedene Konfigurationsmöglichkeiten des Tools, die für die Bedeutungsauflösung relevant sind. Die beiden Hauptkonfigurationsmöglichkeiten „ppr“ und „ppr_w2w“ verändern die Art und Weise mit der der personalisierte PageRank von Wörtern berechnet wird. Beide Varianten verbinden dabei zunächst die zu disambiguierenden Wörter des Eingabetextes mit ihren möglichen Bedeutungen innerhalb des Graphen. Die Varianten unterscheiden sich jedoch in der initialen Gewichtung bei der Berechnung des personalisierten PageRanks. Während die Berechnungsmethode

„ppr“ zunächst das Gewicht auf die möglichen Wortbedeutungen eines Wortes legt, wird bei der Berechnungsmethode „ppr_w2w“ das Gewicht eher auf die möglichen Wortbedeutungen der Wörter im Umgebungskontext gelegt. Somit spielt der umgebende Kontext bei der Konfiguration „ppr_w2w“ eine größere Rolle. Die Autoren beschreiben, dass die Methode „ppr_w2w“ als Verbesserung der Methode „ppr“ entwickelt wurde, um zu verhindern, dass zwei semantisch verwandte Wortbedeutungen eines Wortes sich gegenseitig verstärken und somit der Kontext bei der Bedeutungsauflösung außer Acht gelassen wird. Obwohl die Ergebnisse diese Verbesserung auch untermauern, kann nicht mit Sicherheit gesagt werden, dass dies auch bei einer anderen Wissensquelle wie DBpedia der Fall ist. Deshalb sollten beide Fälle für den Entwurf in Betracht gezogen werden.

Des Weiteren erlauben die Autoren die freie Wahl bei der Größe des Eingabekontextes. Sie beschreiben jedoch, dass einzelne Sätze häufig zu kurz sind, um präzise Ergebnisse bei der Bedeutungsauflösung zu erzeugen. Es wird deshalb empfohlen auch die vorhergehenden und nachfolgenden Sätze ebenfalls zu berücksichtigen [ALS18]. Im Kontext dieser Arbeit sollten deshalb auch beide Varianten in Betracht gezogen und evaluiert werden, um zu prüfen, ob ein größerer Kontext auch bei der Bedeutungsauflösung mit DBpedia bessere Ergebnisse erzielt.

5.8 Entwurf

Nachdem die einzelnen Aspekte der MWA-Erkennung und der Bedeutungsauflösung bei MWAs näher analysiert wurden, wird nun anhand der Ergebnisse der Entwurf eines entsprechenden Systems näher beschrieben. INDIRECT verwendet für die Darstellung und Bearbeitung von Anforderungen eine Graphenstruktur. Hierfür wird ein Framework aus Agenten bereitgestellt, die jeweils Zugriff auf einen zentralen Graphen besitzen und diesen nacheinander bearbeiten. Das entworfene System wird dementsprechend ebenfalls das Framework nutzen, um eine Menge von Agenten bereitzustellen, die die Aufgabe lösen.

5.8.1 Agent für die MWA - Erkennung

Das zweite Teilziel dieser Arbeit ist die Erkennung von MWAs in Anforderungen. Für das Erreichen dieses Teilziels soll nun ein eigenständiger Agent für die MWA-Erkennung entwickelt werden. Dieser soll auf den zentralen Graphen von INDIRECT zugreifen, die entsprechenden Anforderungen herauslesen und darin die vorhandenen MWAs erkennen. Die Analyse in Abschnitt 5.4 hat ergeben, dass ein Ansatz, der auf dem maschinellen Lernen basiert, im Kontext dieser Arbeit die beste Wahl darstellt. Im Speziellen orientiert sich der hier verwendete Ansatz an dem Verfahren in „UW-CSE at SemEval-2016 Task 10: Detecting Multiword Expressions and Supersenses using Double-Chained Conditional Random Fields“ [HSL16] bei dem ein doppelt verkettetes Zufallsfeld verwendet wurde. In dieser Arbeit wird lediglich ein einfach verkettetes Zufallsfeld verwendet, welches mit einem Datensatz basierend auf den Anforderungen verschiedener Projekte trainiert wird. Das einfach verkettete Zufallsfeld bekommt dabei als Eingabe die Wörter der Anforderung in richtiger Reihenfolge, die zugehörigen Lemmata und die Wortartmarkierungen. Anhand dieser Informationen werden die MWAs markiert und in eine Ausgabedatei geschrieben.

In Abbildung 5.1 ist ein grober Aufbau des Agenten dargestellt. Zunächst erhält der Agent Zugriff auf den zentralen INDIRECT-Graphen. Es besteht die Möglichkeit das dieser Graph bereits zu diesem Zeitpunkt durch andere Agenten bearbeitet wurde. Damit der Agent für die MWA-Erkennung jedoch arbeiten kann, ist es wichtig, dass die Anforderungen des Projektes im Graphen bereits in tokenisierter Form vorliegen. Zusätzlich werden die Wortartmarkierungen und Lemmata jedes Wortes benötigt. Diese nötigen Voraussetzungen werden zunächst von dem Agenten geprüft. Falls alle Vorbedingungen erfüllt sind,

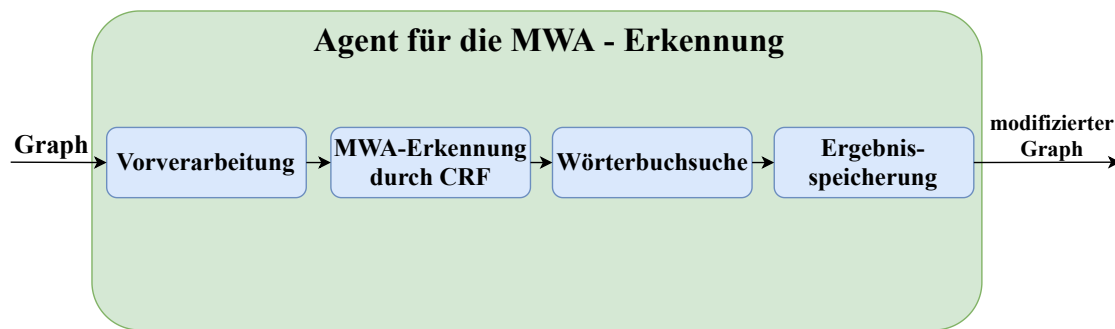


Abbildung 5.1: Agent für die MWA-Erkennung

werden im Rahmen der Vorverarbeitung die Worte jeder enthaltenen Anforderung in korrekter Reihenfolge aus dem Graphen ausgelesen, und mit zusätzlichen Informationen wie Wortartmarkierungen und Lemmata in einer Datei gespeichert. Diese Datei wird anschließend als Eingabe für das einfach verkettete Zufallsfeld verwendet. Das Zufallsfeld erzeugt anschließend eine Ausgabedatei in der alle gefundenen MWAs innerhalb der Anforderung mithilfe des IOB-Formats markiert sind. Basierend auf der Vermutung, dass die bereits in Abschnitt 5.2 erwähnten Randfälle von Adjektiv und Nomen Kombinationen nur selten auftreten und somit auch durch ein einfach verkettetes Zufallsfeld nur schlecht erlernt werden, wird die MWA-Erkennung durch einen weiteren Schritt erweitert. Hierbei sollen alle Kombinationen von Adjektiven und Nomen, die bisher noch nicht von dem Zufallsfeld als Teil eines MWAs markiert wurden, in der verwendeten Wissensquelle gesucht werden. Wird ein solcher Eintrag gefunden, so wird diese Kombination ebenfalls als MWA markiert. Die MWAs werden schlussendlich von dem Agenten aus der Datei ausgelesen und es werden sogenannte „MWA-Token“ für den Graphen erzeugt. Diese MWA-Token beinhalten jeweils den zugehörigen MWA und besitzen Kanten zu den jeweiligen „Token“, die ein Teilwort des MWAs beinhalten.

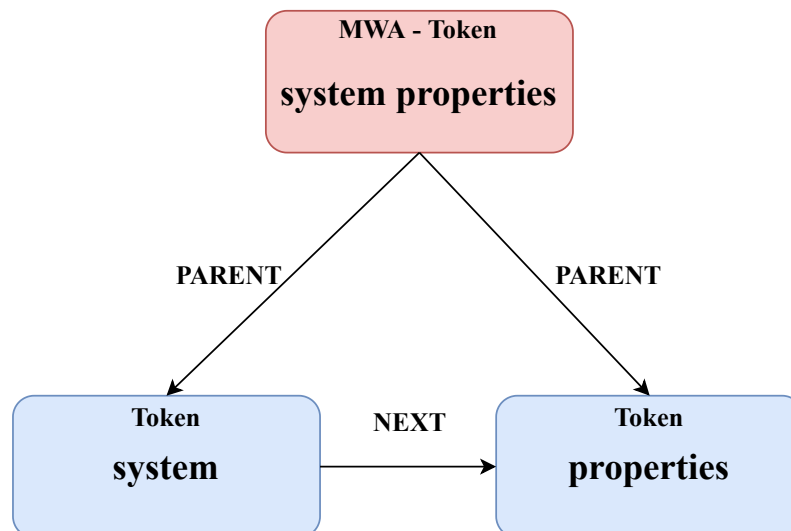


Abbildung 5.2: Tokens nach der Verarbeitung durch den Agenten

In Abbildung 5.2 ist die Graphstruktur nach der Verarbeitung durch den Agenten zu sehen. Als Beispiel sind die Token der Worte „system“ und „properties“ zu sehen, deren Reihenfolge bereits durch andere Agenten mit einer „NEXT“-Kante festgelegt wird. Nachdem der Agent für die MWA - Erkennung die beiden Worte als zusammenhängenden MWA erkannt hat, erstellt er einen MWA-Token, dessen Inhalt aus dem gesamten Ausdruck besteht. Von

dem MWA-Token werden „PARENT“-Kanten zu jedem Token erstellt, das ein Teilwort des entsprechenden MWAs darstellt.

5.8.2 Agent für die Bedeutungsauflösung

Das dritte Teilziel dieser Arbeit ist die Bedeutungsauflösung bei Anforderungen mit Berücksichtigung von MWAs. Für das Erreichen dieses Teilziels soll ebenfalls ein eigenständiger Agent erstellt werden, der die, durch den Erkennungsagenten erstellten MWA-Token, verwendet. Der Graph soll zu diesem Zeitpunkt die in Abbildung 5.2 dargestellte Form besitzen.

Durch die zuvor erstellte Analyse wurde festgestellt, dass im Kontext dieser Arbeit ein wissensbasierter Ansatz die beste Wahl für die Bedeutungsauflösung darstellt. Wie bereits in Abschnitt 5.6 näher analysiert wurde, soll UKB in Kombination mit der Wissensbasis DBpedia für die wissensbasierte Bedeutungsauflösung verwendet werden. Hierfür werden sowohl aus den Datensätzen der Wissensbasis DBpedia, als auch aus den Datensätzen der Wissensbasis WordNet 3.1, Graphen erstellt, die UKB für die Bedeutungsauflösung verwenden kann. Die entsprechenden DBpedia Datensätze wurden hierfür bereits in Abschnitt 5.6.1 analysiert.

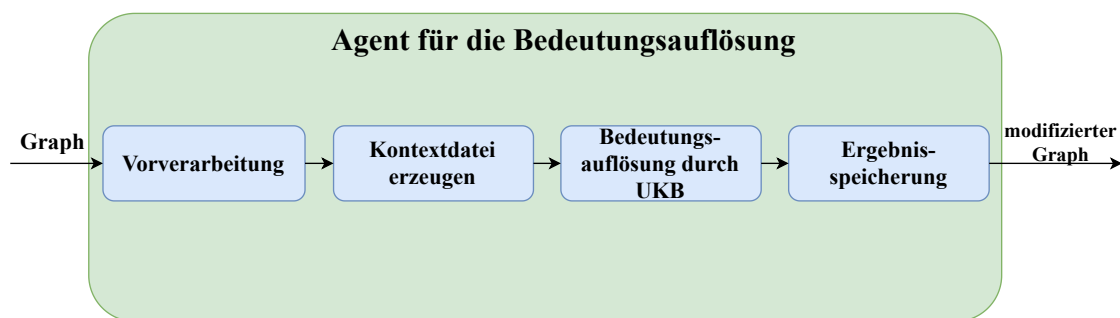


Abbildung 5.3: Agent für die Bedeutungsauflösung

In Abbildung 5.3 ist der grobe Ablauf der Bedeutungsauflösung des Agenten dargestellt. Zunächst wird dem Agenten der Graph mit den zuvor erstellten MWA-Token übergeben. Ähnlich wie bei dem Agenten für die MWA-Erkennung muss zunächst sichergestellt werden, dass die nötigen Vorbedingungen erfüllt sind. Im Fall des Agenten für die Bedeutungsauflösung ist es wichtig, dass bereits der Agent für die MWA-Erkennung und der Abhängigkeitsparser des INDIRECT-Projektes, die Token des Graphen bearbeitet haben. Nachdem der Agent diese Vorbedingung prüft, führt dieser zunächst einige Vorverarbeitungsschritte durch, die für die Bedeutungsauflösung benötigt werden. Dabei annotiert der Agent alle Token und MWA-Token des Graphen mit einem Suchwort, welches für die Suche nach einer Bedeutung in den Wissensbasen verwendet wird. Bei einzelnen Wörtern entspricht dieses Suchwort lediglich dem Lemma des Wortes. Bei den MWAs hingegen wird eine Heuristik angewendet, die auf den Ergebnissen der Analyse in Abschnitt 5.6.2 aufbaut. Als initiales Suchwort wird die Konkatenation der Lemmata der einzelnen Teilworte des MWAs verwendet. Wird keine Definition für den gesamten MWA in der Wissensbasis gefunden, wird zunächst der Kopf des Ausdrucks anhand der Abhängigkeitsbäume bestimmt. Das Suchwort wird anschließend verkleinert, indem immer das am weitesten links liegende Wort ausgehend vom Kopf des Ausdrucks von der Suche ausgeschlossen wird. Dies wird fortgeführt bis ein passender Eintrag in der Wissensbasis gefunden wird.

Beispiel 5.10: Beispiel für die Heuristik bei MWAs

Zielausdruck: „repair facility rating“

Kopf des Ausdrucks: rating

1. Suchwort: repair facility rating
2. Suchwort: facility rating
3. Suchwort: rating

Beispiel 5.10 zeigt einen beispielhaften Ablauf der Heuristik bei dem MWA „repair facility rating“. Zunächst wird versucht eine Bedeutung für den Ausdruck „repair facility rating“ zu finden. Ist dies nicht möglich, so wird der Kopf des Ausdrucks bestimmt, was in diesem Fall dem Teilwort „rating“ entspricht. Ausgehen davon wird immer das am weitesten links liegende Teilwort aus der Suche ausgeschlossen. Wird überhaupt kein Eintrag gefunden, so wird der Ausdruck nicht bei der Bedeutungsauflösung in Betracht gezogen.

Im letzten Schritt der Vorverarbeitung werden die Token und MWA-Token des Graphen in korrekter Reihenfolge ausgelesen und deren Suchwort in eine Kontextdatei geschrieben, die von UKB verwendet wird. Wichtig ist dabei, dass ein Token nur dann in die Kontextdatei geschrieben wird, wenn es kein MWA-Token als Elternknoten besitzt. UKB erzeugt mit der Kontextdatei anschließend eine Ausgabedatei, die die entsprechenden Bedeutungen der MWAs und der einzelnen Wörter der Anforderungen enthält. Diese Ausgabedatei wird anschließend ausgelesen und für jede Bedeutung eines MWAs, wird dem entsprechenden MWA-Token des Graphen ein Attribut mit der Bedeutung angefügt. Das gleiche Verfahren wird mit den Bedeutungen einzelner Wörter verwendet, nur wird bei diesen die Bedeutung an die normalen Token angefügt.

6 Implementierung

Im nachfolgenden Kapitel wird die Implementierung und die hierzu verwendeten Werkzeuge, des in Kapitel 5 beschriebenen Entwurfs näher beschrieben.

6.1 Identifikation von MWAs basierend auf einem bedingten Zufallsfeld

In Abschnitt 5.4 wurde bereits festgelegt, dass ein Ansatz, der auf dem maschinellen Lernen basiert, für die MWA-Erkennung verwendet werden soll. Zur Umsetzung dieses Ansatzes dient die Arbeit „UW-CSE at SemEval-2016 Task 10: Detecting Multiword Expressions and Supersenses using Double-Chained Conditional Random Field“ [HSL16] als Orientierung. Die Autoren verwenden für die SemEval-2016 ein doppelt verkettetes Zufallsfeld für die simultane Erkennung von MWAs und deren semantische Kategorie. Da in dieser Arbeit lediglich die Erkennung von MWAs relevant ist, soll nur ein einfach verkettetes Zufallsfeld mithilfe eines annotierten Datensatzes trainiert werden.

Zum Training eines solchen Zufallsfeldes wird das externe Toolkit „mwetoolkit3“ [Ram12, Ram15] verwendet. Das mwetoolkit3 bietet eine Sammlung von Pythonskripten, die sowohl regelbasierte MWA-Erkennung, als auch MWA-Erkennung basierend auf einem einfach verketteten Zufallsfeld, ermöglichen. Für den regelbasierten Ansatz wird eine Suche nach POS-Tag Mustern innerhalb eines Eingabetextes verwendet. Hierfür muss eine Menge vordefinierter Muster in einer XML-Datei bereitgestellt werden. Anschließend werden diese Muster von dem Toolkit genutzt, um eine Menge von MWA-Kandidaten aus einem Eingabetext zu filtern. Die erstellte Liste kann anschließend zur MWA-Erkennung erneut auf den Eingabetext abgebildet werden. Des Weiteren stellt das Toolkit die Möglichkeit bereit ein einfach verkettetes Zufallsfeld mithilfe eines annotierten Datensatzes zu trainieren. Hierfür muss ein Trainingsdatensatz in dem sogenannten, „DiMSUM-Format“ [noa20, SHJC16], welches auch bei der SemEval-2016 verwendet wurde, erstellt werden.

Das DiMSUM-Format verwendet 17 universelle Wortartmarkierungen. Die Autoren stellen hierfür eine Abbildung des Penn Treebank POS Tagsets auf diese 17 Wortartmarkierungen bereit. Diese Abbildung wird in leicht abgewandelter Form in dieser Arbeit übernommen. Tabelle 6.1 zeigt eine Abbildung der Wortartmarkierungen des DiMSUM-Formats auf die Wortartmarkierungen des Penn Treebank POS Tagsets. Diese Abbildung ist nötig, da INDIRECT für die Wortartmarkierung der Token das Penn Treebank POS Tagset verwendet, das einfach verkettete Zufallsfeld, aber mit den Wortartmarkierungen des DiMSUM-Formats arbeitet.

Tabelle 6.1: Abbildung des Penn Treebank POS Tagsets auf die 17 Wortartmarkierungen [dim]

Wortartmarkierung	Penn Treebank POS Tags
ADJ	JJ, JJR, JJS
ADP	RP, IN
ADV	RB, RBR, RBS, WRB
AUX	MD
CONJ	CC
DET	DT, PDT, WDT
INTJ	UH
NOUN	NN, NNS
NUM	CD
PART	POS, TO
PRON	PRP, PRP\$, WP, WP\$, EX
PROPN	NNP, NNPS
PUNCT	“ -LRB- -RRB- . : HYPH NFP
SYM	NFP, #, \$, SYM, %
VERB	VB, VBP, VBZ, VBD, VBG, VBN
X	FW, LS, XX, ADD, AFX, GW

6.1.1 Trainingsdatensatz

Da das zu implementierende System vor allem auf die Identifikation von MWAs innerhalb von Anforderungsspezifikationen spezialisiert sein soll, wird der verwendete Trainingsdatensatz aus einer Menge von Anforderungsbeispielen zusammengestellt. Hierfür wurden Anforderungen aus den drei Projekten CM1 [HDS06], EBT [CHCC03], GANNT [HHD09] von CoEST [coe] und dem NFR-Datensatz [CHMLP07] extrahiert und anschließend manuell annotiert. In Tabelle 6.2 ist ein Auszug aus dem erstellten Trainingsdatensatz für das einfach verkettete Zufallsfeld zu sehen. Die MWAs innerhalb des Datensatzes sind nach dem IOB-Format annotiert. Jedes isolierte Wort bekommt das Zeichen „O“, das erste Wort eines MWAs bekommt das Zeichen „B“ und ein Wort innerhalb eines MWAs wird mit „I“ annotiert. Für den Bereich zwischen Bestandteilen eines unzusammenhängenden MWAs wird dieselbe Annotation in Form von Kleinbuchstaben durchgeführt.

Tabelle 6.2: Auszug aus dem erstellten Trainingsdatensatz im Dimsum-Format

Position	Wort	Lemma	POS-Tag	IOB	Position	Satznummer
1	The	the	DET	O	0	198
2	human	human	ADJ	B	0	198
3	resource	resource	NOUN	I	2	198
4	class	class	NOUN	I	3	198
5	contains	contain	VERB	O	0	198
6	the	the	DET	O	0	198
7	variable	variable	NOUN	O	0	198
8	holidays	holiday	NOUN	O	0	198
9	.	.	PUNCT	O	0	198

Das mwetoolkit3 verwendet anschließend den annotierten Datensatz, um ein einfach verkettetes Zufallsfeld zur Identifikation von MWAs zu trainieren und in Form einer CRF.model

Datei zu speichern. Nachdem das Training abgeschlossen ist, kann das mwetoolkit3 eine Eingabedatei im DiMSUM-Format annehmen und die MWAs mit dem IOB-Format annotieren. Die IOB Felder und die Position des Vorgängers jedes Wortes müssen in der Eingabedatei die Werte „O“ bzw. „0“ besitzen, damit eine korrekte Annotation möglich ist.

6.1.2 Agent für die MWA-Detektion

Wie bereits zuvor beschrieben stellt INDIRECT ein Framework aus Agenten bereit, die jeweils eine spezifische Aufgabe im System übernehmen. Hierfür existiert die Klasse „AbstractAgent“, die die Methoden „init()“ zur Initialisierung des Agenten und die Methode „exec()“ zum Ausführen der spezifischen Aufgabe, bereitstellt.

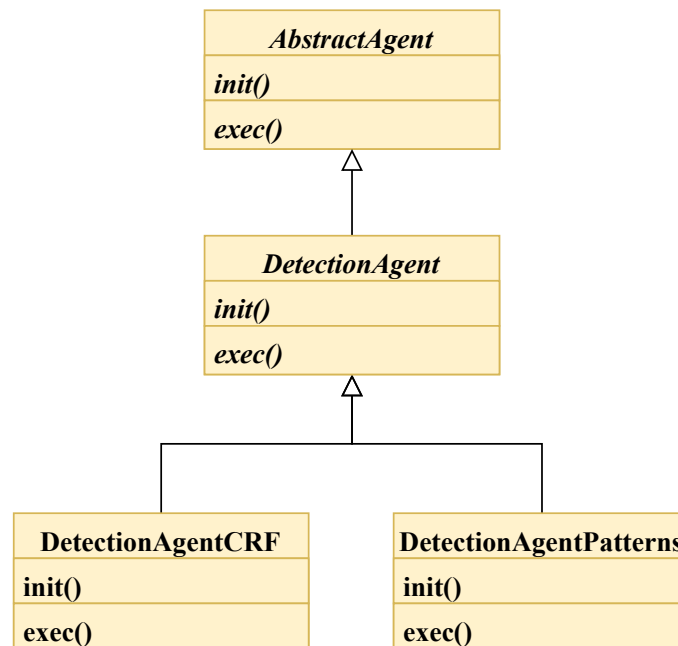


Abbildung 6.1: Klassendiagramm der Agenten für die MWA-Erkennung

In Abbildung 6.1 ist das Klassendiagramm der Agenten, die für die MWA-Erkennung implementiert werden zu sehen. Die Klasse DetectionAgent erbt von der Klasse AbstractAgent und implementiert den allgemeinen Ablauf der MWA-Erkennung. Der „DetectionAgent“ besitzt Zugriff auf den zentralen Graphen von INDIRECT und prüft zunächst, ob der Graph bereits Token enthält, die sowohl lemmatisiert, als auch mit Wortarten markiert wurden. Dies sind die Vorbedingungen, die nötig sind, damit der Agent eine MWA-Erkennung durchführen kann. INDIRECT stellt hierfür eine Vorverarbeitungs-pipeline bereit, die jede Anforderung tokenisiert und grundlegende Vorverarbeitungsschritte wie das Erzeugen von Lemmata und die Annotation mit Wortarten durchführt. Nachdem die Vorbedingungen geprüft wurden, iteriert der Agent über alle Anforderungen, die bereits im zentralen INDIRECT Graphen vorhanden sind. Für jede dieser Anforderungen werden die Token, so wie deren Lemmata und Wortarten extrahiert und anschließend eine Eingabedatei „inputFile.txt“ im DiMSUM-Format erzeugt. Diese Eingabedatei kann anschließend von dem mwetoolkit3 genutzt werden, um eine MWA-Erkennung durchzuführen. Hierfür existieren die Klassen DetectionAgentCRF und DetectionAgentPatterns, die jeweils von der Klasse DetectionAgent erben. Die Klasse DetectionAgentCRF verwendet das zuvor trainierte einfach verkettete Zufallsfeld, um die MWAs in der Eingabedatei zu annotieren. Die Klasse DetectionAgentPatterns hingegen verwendet die regelbasierte MWA-Erkennung basierend auf Mustern in einer XML-Datei. Beide Agenten erzeugen eine Ausgabe, die in

der Datei „outputFile.txt“ gespeichert wird. Dabei ist wichtig zu erwähnen, dass die Klasse `DetectionAgentPatterns`, die eine regelbasierte MWA-Erkennung durchführt, lediglich als Zusatz implementiert wurde, da das `mwetoolkit3` diese Funktion anbietet. Die regelbasierte MWA-Erkennung ist eigentlich nicht Bestandteil dieser Arbeit und wird deshalb auch nicht weiter ausgeführt oder evaluiert. Nachdem die Ausgabedatei durch das `mwetoolkit3` erzeugt wurde, erweitert die Klasse `DetectionAgent` die Suche nach MWAs mithilfe einer Wörterbuchdatei der Wissensbasis DBpedia, die im Rahmen der Bedeutungsauflösung näher erläutert wird. Hierfür werden Einträge zu allen Adjektiv+Nomen Wortkombinationen, die noch nicht durch das Zufallfeld oder die Muster markiert wurden, in der Wörterbuchdatei gesucht. Ist ein Eintrag vorhanden, so wird diese Wortkombination ebenfalls als MWA in der Ausgabedatei markiert. Schlussendlich wird die Ausgabedatei ausgelesen und für jeden gefundenen MWA wird nun ein neues sogenanntes „MWA-Token“ erzeugt. Dieses Token enthält den gesamten MWA als Inhalt und besitzt Kanten zu den Token, die die einzelnen Wörter des MWAs enthalten. Abbildung 6.2 ist eine beispielhafte Darstellung

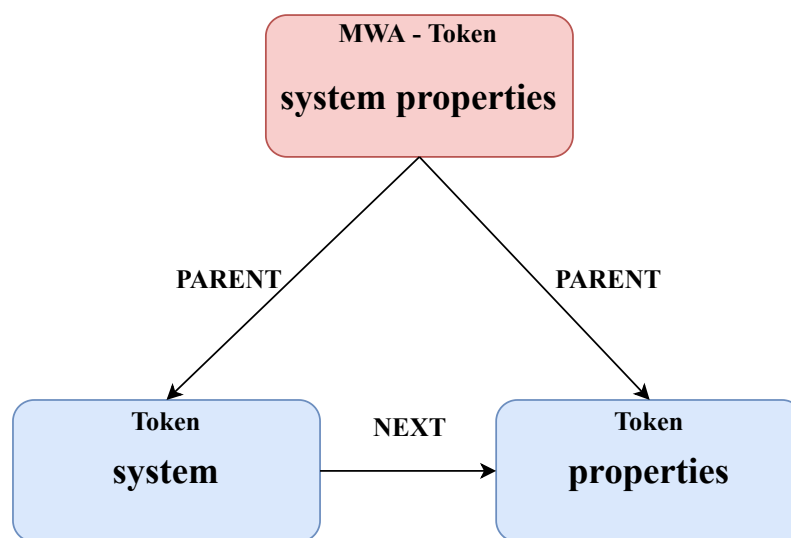


Abbildung 6.2: MWA-Token mit Kindern

eines erstellten MWA-Token nachdem der Agent die MWA-Erkennung abgeschlossen hat. Das MWA-Token besitzt Kanten mit dem Bezeichner „PARENT“ zu allen Kindtoken, die ein Teilwort des MWAs beinhalten.

6.2 Wissensbasierte Bedeutungsauflösung

Für die wissensbasierte Bedeutungsauflösung mit Berücksichtigung von MWAs wird die bereits erwähnte Programmsammlung „UKB“ [AS09, aso20] verwendet. UKB ist eine Sammlung von C++ Programmen und stellt eine Menge von Werkzeugen zur wissensbasierten Bedeutungsauflösung anhand der Berechnung eines personalisierten PageRanks bereit. Wie bereits erwähnt ist UKB nicht auf eine einzige Wissensquelle beschränkt, sondern arbeitet auf jedem beliebigen Graphen, der dem Tool zur Verfügung gestellt wird. Wichtig ist jedoch, dass der Graph eine Struktur besitzen muss, die bei der Berechnung des personalisierten PageRanks relevante Ergebnisse für die Bedeutungsauflösung erzeugt. In Abschnitt 5.6.1 wurde deshalb bereits analysiert, welche Datensätze der Wissensbasis DBpedia sich zur Erstellung eines solchen Graphen eignen. Dabei wurde festgestellt, dass das „Linked Hypernyms Dataset (LHD)“ und das „categories dataset“ einen Graphen induzieren, der eine Hierarchie von Ober- und Unterkonzepten darstellt und dieser sich somit aufgrund der Ähnlichkeit zum WordNet Graphen für die Bedeutungsauflösung eignen sollte. Die aktuellen Versionen dieser Datensätze werden der aktuellen Version von

DBpedia entnommen [DBp]. Die RDF Aussagen des Hypernym Datensatzes stellen dabei Relationen dar, die jeweils einen DBpedia Eintrag zu seinem entsprechenden Hypernym zuordnen. Der Kategorien Datensatz stellt dagegen eine Menge von RDF Aussagen bereit, die einen DBpedia Eintrag zu einer Menge von Kategorien zuordnen. Die Relationen, die durch diese RDF Aussagen dargestellt werden, können bereits als Graph interpretiert werden. Hierfür werden alle Subjekte und Objekte der RDF Aussagen als die Knoten des Graphen angesehen. Die Zuordnungen, die durch die RDF Aussagen modelliert werden, repräsentieren dabei die Kanten des Graphen. Verwendet man die DBpedia Datensätze in dieser Weise, so muss beachtet werden, dass Überschneidungen nicht doppelt in den Graphen aufgenommen werden. Der Hypernym Datensatz ordnet jedoch immer nur DBpedia Einträge zu anderen Einträgen oder zu Typen der DBpedia Ontologie zu. Der Kategorien Datensatz besitzt hingegen nur Relationen zwischen DBpedia Einträgen und Kategorien bzw. Relationen zwischen den Kategorien, weshalb keine Überschneidungen existieren. Die RDF Aussagen müssen nun in eine Form gebracht werden, die von UKB verarbeitet werden kann. Hierfür benötigt UKB eine Textdatei, die den zu verwendenden Graphen repräsentiert, indem pro Zeile ein Knoten *u* zu einem Knoten *v* zugeordnet wird.

Beispiel 6.1: Beispiel für die Umwandlung der Datensätze zum UKB-Graphen

RDF Aussage aus dem Hypernym Datensatz:

```
< http://dbpedia.org/resource/Tebra_castle >
```

```
< http://www.w3.org/1999/02/22-rdf-syntax-ns#type >
```

```
< http://dbpedia.org/ontology/Castle > .
```

Entsprechende Zeile in der Textdatei, die den Graphen repräsentiert:

```
u:Tebra_castle v:Castle
```

In Beispiel 6.1 ist ein Beispiel für die Umwandlung einer RDF Aussage in eine Zeile, die von UKB verarbeitet werden kann. „Tebra_castle“ ist ein Knoten *u*, der mit dem Knoten *v* „Castle“ verbunden ist. „Castle“ stellt in diesem Fall das Hypernym dar. Auf diese Art und Weise wird aus dem Hypernym und dem Kategorien Datensatz von DBpedia eine Textdatei erstellt. Anschließend wird diese von UKB in eine Binärdatei umgewandelt, die später für die Bedeutungsauflösung verwendet wird. Aufgrund der großen Menge an Daten, die in den Datensätzen vorhanden ist, ist die entsprechende Binärdatei des Graphen sehr groß, wodurch die Verarbeitungszeit bei der Bedeutungsauflösung steigt. Um eine schnellere Alternative zu bieten, wurde deshalb auch ein Graph erzeugt, der lediglich aus den Relationen des „Linked Hypernym Dataset“ besteht.

Wie bereits in der Analyse erwähnt eignet sich eine Wissensbasis wie DBpedia lediglich für eine Bedeutungsauflösung bei Nomen und Eigennamen. Um sowohl die Möglichkeit zu bieten dennoch die anderen Wortarten zu disambiguieren, als auch um einen Vergleich bei der Evaluation zu besitzen, wird zusätzlich noch ein Graph der Wissensbasis WordNet 3.1 erstellt. UKB stellt hierfür ein Skript bereit, das diese Graphendatei aus den Datenbankdateien von WordNet 3.1 erstellt.

Die Graphendatei alleine ist jedoch nicht ausreichend, um eine Bedeutungsauflösung bei Eingabetexten durchzuführen, da UKB bei einer Eingabe noch nicht weiß zu welchen Konzepten des Graphen die entsprechenden Wörter zugeordnet werden müssen. Hierfür wird eine weitere Textdatei benötigt, die sogenannte „Wörterbuchdatei“, die englischen Begriffen ein oder mehrere mögliche Konzepte des verwendeten Graphen zuordnet. Diese Konzepte werden von UKB dann als mögliche Bedeutungen für den Begriff verwendet.

Für die Wissensbasis Wordnet 3.1, stellt UKB ein Skript bereit, das aus den vorhandenen Datenbankdateien eine solche Datei erstellt. Für die Wissensbasis DBpedia hingegen, muss diese Datei manuell aus den vorhandenen Datensätzen erstellt werden. Hierbei können mehrere Fälle unterschieden werden.

Beispiel 6.2: Fall 1: Einträge in der Wörterbuchdatei für Wörter mit mehreren Bedeutungen

Digraph Digraph_(orthography) Digraphs_and_trigraphs Directed_graph Digraphia

Beispiel 6.2 zeigt einen Eintrag für ein Wort, welches in DBpedia mehrere mögliche Bedeutungen besitzt. Dem Wort „Digraph“, wird durch den Eintrag die möglichen Bedeutungen „Digraph_(orthography)“, „Digraphs_and_trigraphs“, „Directed_graph“ und „Digraphia“ zugeordnet. Umgesetzt wird dies mithilfe des „disambiguations“ Datensatz, der jedem mehrdeutigen Ausdruck die möglichen DBpedia Einträge zuordnet. Jeder mehrdeutige Eintrag innerhalb dieses Datensatzes erhält dann eine Zeile in der Wörterbuchdatei, wie sie in Beispiel 6.2 zu sehen ist.

Beispiel 6.3: Fall 2: Einträge in der Wörterbuchdatei für Wörter deren Schreibweise nicht in der Wissensquelle vorhanden ist

AfghanistanMilitary Afghan_Armed_Forces

Beispiel 6.3 zeigt den zweiten möglichen Fall bei dem ein Wort, dessen Schreibweise nicht mit einem Eintrag in der Wissensquelle übereinstimmt. Das Wort „AfghanistanMilitary“ ist aufgrund der Schreibweise nicht direkt mit einem DBpedia Eintrag verknüpfbar. Jedoch entspricht die Bedeutung des Wortes sinngemäß dem Eintrag „Afghan_Armed_Forces“. Damit dies von UKB erkannt wird, wird der in Beispiel 6.3 gezeigte Eintrag in der Wörterbuchdatei eingefügt. Umgesetzt wird dies mit dem „redirects“ Datensatz, der eine Menge von N-Triplen bereitstellt, die Begriffe mit ähnlicher, aber nicht direkt passender Schreibweise, auf die passenden DBpedia Einträge umleiten.

Beispiel 6.4: Fall 3: Eindeutige Begriffe

Initialization_(programming) Initialization_(programming)

Beispiel 6.4 zeigt den letzten Fall, bei dem ein Begriff eindeutig zu einem DBpedia Eintrag zugeordnet werden kann. In DBpedia ist für den Begriff „Initialization_(programming)“ ein Eintrag vorhanden, welcher denselben Namen verwendet. Dies ist eine eindeutige Zuordnung und wird in der Wörterbuchdatei durch eine Selbstabbildung dargestellt.

Eine weitere wichtige Funktion, die UKB zur Berechnung des personalisierten PageRanks bereitstellt, ist die Verwendung von Gewichten bei der Verlinkung von Begriffen zu mehreren Konzepten der Wissensbasis. Diese Gewichte können nach belieben festgelegt werden und die Gewichte aller möglichen Bedeutungen eines Wortes werden bei der Berechnung des personalisierten PageRanks normalisiert, so dass diese sich zu dem Wert 1 aufsummieren. Die Autoren von UKB beschreiben, dass die Gewichte in der Wörterbuchdatei zur Wissensbasis WordNet, der absoluten Häufigkeit der Bedeutung in mehreren Dokumenten entsprechen [aso20]. Bei der manuellen Erstellung der Wörterbuchdatei zur Wissensbasis DBpedia wird deshalb ein ähnliches Vorgehen angewandt. Um eine Einschätzung dafür zu

bekommen, wie häufig bestimmte DBpedia Einträge auftreten, wird der „wikilinks“ Datensatz verwendet. Die darin enthaltenen RDF Aussagen stellen Zuordnungen zwischen Wikipedia Einträgen dar. Ein Wikipedia Eintrag besitzt eine Relation zu allen anderen Wikipedia Einträgen, zu denen einer einen Link in seinem Text enthält. Zählt man wie häufig ein Wikipedia Eintrag in Form eines Links in allen anderen Wikipedia Einträgen auftritt, so ist diese Anzahl gleichbedeutend mit der absoluten Häufigkeit der Bedeutung im gesamten Wikipedia Korpus.

Beispiel 6.5: Auszug aus der Wörterbuchdatei mit Gewichten

Digraph Digraph_(orthography):544 Digraphs_and_trigraphs:73 Directed_graph:593
Digraphia:25

In Beispiel 6.5 ist erneut der Auszug aus der Wörterbuchdatei zu sehen, der schon in Beispiel 6.2 aufgezeigt wurde. In diesem Auszug sind jedoch bereits die absoluten Häufigkeiten der einzelnen Bedeutungen im Wikipedia Korpus eingetragen. Diese Häufigkeiten werden bei der Berechnung durch UKB normalisiert, so dass sie sich insgesamt zu dem Wert 1 aufsummieren und schließlich werden die normalisierten Häufigkeiten bei der Berechnung des personalisierten PageRanks verwendet.

6.2.1 Agent für die Bedeutungsauflösung

Ähnlich wie auch bereits bei der MWA-Detektion in Abschnitt 6.1.2 beschrieben, wird für die Bedeutungsauflösung ebenfalls ein Agent für das INDIRECT Framework implementiert.

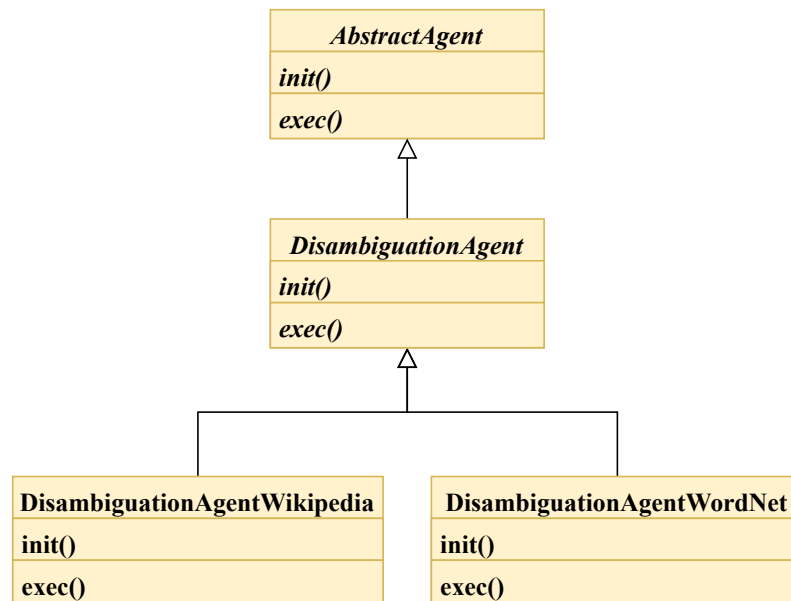


Abbildung 6.3: Klassendiagramm der Agenten für die Bedeutungsauflösung

Abbildung 6.3 zeigt das Klassendiagramm zu den erstellten Agenten für das INDIRECT Framework. Es wird eine abstrakte Klasse „DisambiguationAgent“ implementiert, die von der Klasse „AbstractAgent“ erbt. Sie enthält das Grundgerüst des Algorithmus für die Bedeutungsauflösung, der später vom System verwendet wird. Die beiden Klassen „DisambiguationAgentWikipedia“ und „DisambiguationAgentWordNet“ enthalten die nötigen Bestandteile zur Nutzung von DBpedia bzw. WordNet als Wissensquelle. Zunächst

überprüft der Agent, ob alle nötigen Vorbedingungen erfüllt sind. Wichtig ist hierbei vor allem, dass sowohl der Agent für die MWA-Erkennung, als auch der Agent für die Erstellung von Abhängigkeitsbäumen den Graphen bereits bearbeitet haben. Der Agent für die Bedeutungsauflösung benötigt nämlich deren Informationen, um eine korrekte Bedeutungsauflösung durchführen zu können. Bevor der Agent die einzelnen Anforderungen des INDIRECT-Graphen bearbeitet, erhält jedes Token, abhängig davon welcher Agent gerade den Graphen bearbeitet, ein Attribut „wikipedia_tag“ bzw. „wordnet_tag“. Dieses Attribut stellt den Begriff dar mit dem das Token in UKB repräsentiert wird und wird somit auch als Suchbegriff für eine passende Bedeutung verwendet

Beispiel 6.6: Beispieltags für die Token und MWA-Token

- value: systems, wikipedia_tag: System
- value: program administrators, wikipedia_tag: Administrator
- value: spare parts, wordnet_tag: part

Beispiel 6.6 zeigt einige Beispiele für die Tags die von beiden Agenten vor der Bedeutungsauflösung vergeben werden. Für einfache Token die lediglich ein einziges Wort enthalten, wird der Tag erstellt, indem das Wort lemmatisiert wird. Für alle MWA-Tokens, werden zunächst die Lemmata der einzelnen Bestandteile durch ein „-“ miteinander verbunden. Die Suchbegriffe für die Wissensquelle DBpedia beginnen außerdem immer mit einem Großbuchstaben, was daran liegt, dass alle Konzepte des Graphen mit eine Großbuchstaben anfangen. Die WordNet-Konzepte hingegen sind immer vollständig klein geschrieben.

Anschließend wird in der entsprechenden Wörterbuchdatei der Wissensquelle nach dem erstellten Tag durchsucht. Wird ein passender Eintrag gefunden, so muss nichts verändert werden. Ist kein entsprechender Eintrag im Wörterbuch vorhanden, so wird die bereits im Entwurf verwendete Heuristik angewandt. Dabei wird zunächst der Abhängigkeitsbaum des MWAs untersucht, um den Kopf des Ausdrucks zu identifizieren. Nun wird immer zunächst das Lemma, welches am weitesten Links vom Kopf liegt von der Suche im Wörterbuch ausgeschlossen. Dieses Verfahren wird fortgeführt bis ein Eintrag gefunden wurde, oder keine Wörter mehr für die Suche verblieben sind. Nachdem die Wörterbuchtags an die Token und MWA-Token angehängt wurden, kann der Agent mit der Bedeutungsauflösung beginnen. Hierfür iteriert dieser zunächst über alle Anforderungen, die in dem zentralen Graphen von INDIRECT enthalten sind. Für jede Anforderung werden sowohl die Token, als auch die MWA-Token extrahiert und es wird eine Kontextdatei erstellt, die UKB für die Bedeutungsauflösung verwendet.

Beispiel 6.7: Beispiel für eine Kontextdatei

```
ctx_0
Software_error##35#1 Error##95#1 Loss##135#1 Commandability##155#1
Communication##175#1
```

Beispiel 6.7 zeigt wie eine Anforderung in einer Kontextdatei der zur Bedeutungsauflösung mit der Wissensquelle DBpedia dargestellt wird. Der Kontext besitzt folgende Form: Lemma#POS-Tag#Wort-ID#Kontrollnummer. Die Wort-ID ist eine eindeutige Identifikationsnummer für das Wort im vorhandenen Kontext und die Kontrollnummer gibt an, ob das Wort disambiguiert oder lediglich für die Berechnung des PageRanks verwendet werden soll. Die Wortartmarkierung ist optional und wird nur bei der Bedeutungsauflösung mit

WordNet 3.1 verwendet, da der DBpedia Graph nicht zwischen Wortarten unterscheidet.

Beispiel 6.8: Beispiel für eine Kontextdatei

```
ctx_0
software_error#n#35#1 error#n#95#1 loss#n#135#1 commandability#n#155#1
communication#n#175#1
```

Beispiel 6.8 zeigt den Kontext erneut, nur diesmal wurde sie von der Disambiguation-AgentWordNet Klasse erstellt. Wie zu erkennen ist, wurden die Wortartmarkierungen für die einzelnen Nomen verwendet.

Die Größe des verwendeten Kontexts ist frei wählbar, jedoch verwenden die implementierten Agenten immer die gesamte Anforderung als Kontext. Die DisambiguationAgent-Wikipedia Klasse verwendet für den Kontext lediglich alle Nomen und Eigennamen einer Anforderung, da nur diese einen möglichen Eintrag in der Wissensquelle DBpedia besitzen. Die DisambiguationAgentWordNet Klasse hingegen verwendet alle Adjektive, Nomen, Adverbien und Verben der Anforderungen für die Bedeutungsauflösung. Anschließend startet der Agent extern UKB, in dem der verwendete Graph, das Wörterbuch und die entsprechende Kontextdatei übergeben wird. UKB erzeugt anschließend eine Ausgabe, in der jedem Wort des Kontexts eine Bedeutung aus der Wissensquelle zugeordnet wird.

Beispiel 6.9: Ausschnitt aus einer Ausgabedatei von UKB

```
ctx_3 64 Error !! Error
ctx_3 94 Location !! Location
ctx_3 164 EEPROM !! EEPROM
ctx_3 214 Watchdog_timer !! Watchdog_Timer
ctx_3 264 Software_bug !! Software_error
```

Beispiel 6.9 zeigt einen Ausschnitt aus einer möglichen Ausgabedatei von UKB, wenn DBpedia als Wissensquelle verwendet wird. Die Ausgabe besitzt die Form: Kontextnummer _Wort-ID_Bedeutung_!!_Wort. Diese Ausgabe wird von den Agenten ausgelesen und in Form eines weiteren Attributs an die Tokens angehängt.

Beispiel 6.10: Bedeutungsattribute eines MWA-Token

```
value: software error,
wikipedia_tag: Software_error,
wordnet_tag: software_error,
wordsense_wikipedia: Software_bug,
wordsense_wordnet: 07314710-n
```

Beispiel 6.10 zeigt ein Beispiel für die zwei Attribute, die nach der Bedeutungsauflösung an die Token angehängt werden. Das Attribut „wordsense_wikipedia“ gibt die Bedeutung

in der Wissensquelle DBpedia an. Die Bedeutung entspricht dem Endstück des Links „http://dbpedia.org/resource/Software_bug“. Das Attribut „wordsense_wordnet“ gibt die Bedeutung in der Wissensquelle WordNet 3.1 an. Die Bedeutung besitzt die Form: Synset-ID - Wortartmarkierung.

7 Evaluation

In diesem Kapitel wird der in Kapitel 5 beschriebene Ansatz und die in Kapitel 6 beschriebene Implementierung anhand verschiedener Techniken evaluiert. Die Evaluation ist in mehrere Stufen aufgeteilt, wobei zunächst für die MWA-Erkennung und die Bedeutungsauflösung eine separate intrinsische Evaluation durchgeführt wird, bei der die Komponenten einzeln betrachtet werden. Anschließend werden die Ergebnisse beider Teilsysteme in Kombination betrachtet. Die Aufteilung der Evaluation in Stufen ist wichtig, um eine differenzierte Interpretation der Ergebnisse zu ermöglichen. Betrachtet man lediglich die kombinierten Ergebnisse, so kann man zwar die Performanz des Systems insgesamt beurteilen, jedoch können mögliche Fehlerquellen bei den einzelnen Teilsystemen schlechter identifiziert werden. Durch die separaten intrinsischen Evaluationen können die Ergebnisse der einzelnen Teilsysteme näher betrachtet und bestimmt werden, ob die Performanz des Systems durch bestimmte Teilsysteme negativ beeinflusst wird.

7.1 Evaluationsdatensatz

Zur Evaluation beider Teilsysteme wird ein Evaluationsdatensatz benötigt, mit dem die Güte beider Teilsysteme getestet wird. Da das entwickelte System vor allem mit Anforderungsspezifikationen arbeitet, muss auch der Datensatz aus einer Menge von Anforderungen erstellt werden, damit eine präzise Einschätzung der Leistung des Systems möglich ist. Dieser Datensatz muss eine Musterlösung, auch Goldstandard genannt, für die Aufgabe beider Teilsysteme enthalten. Dieser Goldstandard wird bei der Evaluation anschließend als Referenz verwendet. Wie bereits in Kapitel 6 beschrieben, arbeitet das Teilsystem für die MWA-Erkennung mit einem einfach verketteten Zufallsfeld, welches mit einem Trainingsdatensatz trainiert wurde. Dieser Trainingsdatensatz ist aus einer Menge von Anforderungen aufgebaut, in denen manuell die MWAs mithilfe des IOB-Formats annotiert wurden. Diese Annotationen stellen somit eine Musterlösung für die MWA-Erkennung dar, weshalb der Trainingsdatensatz sich auch als Evaluationsdatensatz für die MWA-Erkennung eignet. Da der Trainingsdatensatz jedoch keine Musterlösung für die Bedeutungsauflösung enthält, muss diese zunächst manuell hinzugefügt werden, indem darin die Bedeutungen der Nomen, Eigennamen und MWAs manuell annotiert werden.

In Tabelle 7.1 ist ein Auszug aus dem erstellten Evaluationsdatensatz zu sehen. Das DiMSUM-Format, welches bereits beim Training des einfach verketteten Zufallsfeldes verwendet wurde, wurde bei der Bedeutungsannotation beibehalten. Die ersten sieben Spalten der Tabelle wurden bereits in Kapitel 6 näher erläutert. Zusätzlich wurden zwei

Tabelle 7.1: Auszug aus dem Evaluationsdatensatz

Position	Wort	Lemma	POS- Tag	IOB	Position	Satz	DBpedia	WordNet
1	The	the	DET	O	0	29		
2	DPU- TMALI	DPU- TMALI	PROPN	O	0	29	-	-
3	shall	shall	AUX	O	0	29		
4	install	install	VERB	O	0	29		
5	a	a	DET	O	0	29		
6	callback	callback	NOUN	B	0	29	Callback	06594579- - (com- n.0 puter - pro- gram- ming).1
7	routine	routine	NOUN	I	6	29		
8	to	to	PART	O	0	29		
9	respond	respond	VERB	O	0	29		
10	to	to	PART	O	0	29		
11	the	the	DET	O	0	29		
12	DCI	DCI	PROPN	B	0	29	Interrupt	11467628- .0 n.0
13	Error	Error	PROPN	I	12	29		
14	Interrupt	Interrupt	PROPN	I	13	29		
15	.	.	PUNCT	O	0	29		

weitere Spalten eingefügt, die einem Ausdruck seine korrekte Bedeutung in Form von den Einträgen der Wissensquelle DBpedia und den Synset-IDs der Wissensquelle WordNet 3.1 zuordnen. Um einen präzisen Vergleich zwischen WordNet 3.1 und DBpedia zu erhalten, wurden lediglich Wortarten, die in beiden Wissensquellen vorhanden sind, also Nomen, Eigennamen und MWAs, mit einer Bedeutung annotiert. Die Bedeutung eines Ausdrucks wird durch die eindeutige Benennung eines Eintrags in DBpedia repräsentiert. Dies entspricht dem eindeutigen Ende eines „http://dbpedia.org/resource/“ Links. Beispielsweise ist die Bedeutung „Callback_(computer_programming)“ aus dem Link „http://dbpedia.org/resource/Callback_(computer_programming)“ entnommen. Die WordNet 3.1 Bedeutungen entsprechen den Synset-IDs der Synsets, die die Bedeutungen repräsentieren, was bei der sechsten Zeile beispielsweise der Zahl „06594579“ entspricht. Der Anhang „-n“ bei den WordNet Bedeutungen gibt an, ob die Bedeutung einem Nomen, Verb, Adjektiv oder Adverb entspricht. Im Fall eines MWAs erhalten außerdem sowohl die DBpedia, als auch die WordNet Bedeutungen den Anhang „.1“, falls die zugewiesene Bedeutung für den gesamten MWA gilt, oder den Anhang „.0“, falls die zugewiesene Bedeutung nur für einen Teil des MWAs gilt. Für jeden MWA wurde die passende Bedeutung immer in die Zeile des ersten Teilwortes eingetragen. Falls ein Wort keine korrekte Bedeutung in einer der Wissensbasen besitzt, so wurde in die entsprechende Spalte ein „-“ eingetragen. Im Evaluationsdatensatz ist nicht erkennbar, welche Bestandteile eines MWAs zu einer annotierten Teilbedeutung beigetragen haben, da dies für die nachfolgende Evaluation nur eine untergeordnete Rolle spielt. In Kapitel 6 wurde bereits beschrieben, dass das System jedoch für jedes Wort und MWA einen Tag erstellt, der als Wort für die Suche in der entsprechenden Wissensbasis verwendet wird. Dieser Tag entspricht dann der Konkatination der Lemmata der Wörter, die für die Teilbedeutung eine Rolle spielen.

Tabelle 7.2: Statistiken über einzelne Wörter im Evaluationsdatensatz

Projekt	Anforderungen	Wörter	Wörter (a)	K (DB)	ND (DB)	K (WN)	ND (WN)
CM1	31	571	91	43	48	37	54
EBT	41	565	101	84	17	88	13
GANNT	136	2218	450	389	61	395	55
nfr1	33	489	96	78	18	95	1
nfr2	41	515	94	80	14	92	2
nfr3	80	1784	228	173	55	212	16
nfr4	85	1622	179	125	54	170	9
nfr5	90	1095	158	127	31	149	9
nfr6	105	1507	245	214	31	223	22
nfr7	26	351	49	35	14	45	4
nfr8	101	1554	272	249	23	243	29
nfr9	23	403	45	34	11	42	3
nfr10	58	1029	247	210	37	247	0
nfr11	18	359	65	50	15	62	3
nfr12	25	504	104	92	12	87	17
nfr13	38	612	84	66	18	70	14
nfr14	17	299	46	39	7	37	9
nfr15	30	513	73	50	23	63	10
Gesamt	978	15990	2554	2088	466	2294	260

Tabelle 7.2 zeigt verschiedene Statistiken bezüglich der Anzahl der Anforderungen und Wörter pro Projekt. Zusätzlich gibt die Spalte Wörter (a) an, welche einzelnen Wörter eine Annotation im Goldstandard besitzen. Die Spalten K und ND geben die Anzahl der Wörter an, für die eine passende Bedeutung gefunden wurde bzw. für die keine passende Bedeutung gefunden wurde. Ergänzend dazu wurde zwischen den Wissensbasen DBpedia (DB) und WordNet 3.1 (WN) unterschieden. Bei der Anzahl der annotierten Wörter sind die MWAs noch nicht mit einbezogen.

Tabelle 7.3 zeigt verschiedene Statistiken bezüglich der MWAs, die im Evaluationsdatensatz enthalten sind. Aufgelistet werden hierbei die Datensätze deren Anforderungen in der Evaluation verwendet werden, so wie die Anzahl der darin enthaltenen MWAs, die Anzahl der vollständig vorhandenen korrekten Bedeutungen (K), die Anzahl der teilweise korrekten Bedeutungen (TK) und die Anzahl der nicht vorhandenen Bedeutungen (IK). Diese Werte wurden für die beiden verwendeten Wissensquellen DBpedia (DB) und WordNet 3.1 (WN) aufgelistet. Eine Bedeutung wird als teilweise korrekt gewertet, wenn nur für einen Bestandteil des MWAs eine passende Bedeutung gefunden wurde, diese aber dennoch für den gesamten Ausdruck als angemessen befunden wurde. Eine Teilbedeutung wird immer dann als angemessen angesehen, wenn sie eine allgemeinere Form der Entität, die durch den MWA beschrieben wird, darstellt.

Beispiel 7.1:

- MWA: end date
- teilweise korrekte Bedeutung aus DBpedia: Calendar_date

Beispiel 7.1 zeigt einen möglichen Fall, bei dem eine Bedeutung als teilweise korrekt gewertet wird. Für den gesamten MWA „end date“ ist weder in DBpedia, noch in WordNet 3.1, eine passende Bedeutung vorhanden. Jedoch enthalten beide Wissensquellen eine Definiti-

Tabelle 7.3: MWA Statistiken des Evaluationsdatensatzes

Projekt	MWAs	K (DB)	TK (DB)	ND (DB)	K (WN)	TK (WN)	ND (WN)
CM1	51	10	27	14	2	36	13
EBT	46	7	33	6	2	37	7
GANNT	171	20	123	28	8	102	61
nfr1	34	7	20	7	5	24	5
nfr2	43	7	27	9	4	35	4
nfr3	219	23	110	86	14	162	43
nfr4	205	16	129	60	9	186	10
nfr5	114	27	50	37	8	104	2
nfr6	129	55	39	35	26	87	16
nfr7	42	7	23	12	4	36	2
nfr8	127	30	64	33	15	80	32
nfr9	50	8	30	12	2	41	7
nfr10	33	4	25	4	4	27	2
nfr11	27	5	10	12	2	16	9
nfr12	49	20	9	20	5	28	16
nfr13	65	12	42	11	5	49	11
nfr14	23	5	13	5	2	15	6
nfr15	46	6	33	7	3	32	11
Gesamt	1474	269	807	398	120	1097	257

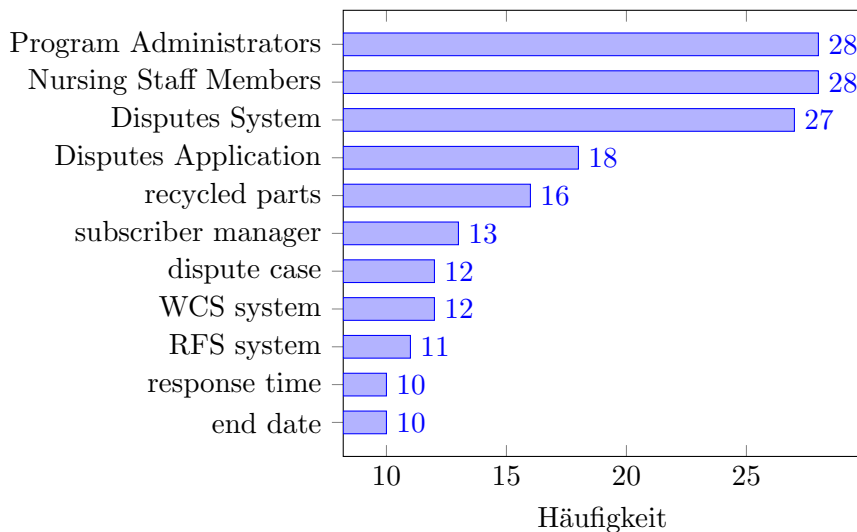


Abbildung 7.1: Häufigste MWAs im Evaluationsdatensatz

on für das Teilwort „date“. Da die Bedeutung des Wortes „date“ lediglich eine allgemeinere Form des MWAs „end date“ darstellt, kann diese genutzt werden, um dennoch einen Teil der Informationen zu erhalten. Sie wird deshalb als angemessen gewertet.

7.1.1 Häufigste MWAs

In Abbildung 7.1 ist ein Graph zu sehen, der die MWAs des Evaluationsdatensatzes anhand ihrer Häufigkeiten auflistet. Deutlich erkennbar ist, dass die häufigsten MWAs vor allem zusammengesetzte Nomen wie die Ausdrücke „Program Administrators“, oder „Nursing Staff Members“ sind. Zusätzlich werden in den Anforderungen auch oft die Eigennamen

der Systeme genannt wie beispielsweise das „RFS system“ oder das „WCS system“.

7.2 Evaluationsmetriken

Bevor die Evaluationsergebnisse aufgeführt und analysiert werden, ist es zunächst wichtig zu klären, welche Metriken bei der Evaluation eine Rolle spielen und wie diese berechnet werden. Der annotierte Datensatz dient als Goldstandard für die Evaluation des Systems. Bei der Auswahl der Metriken zur Evaluation muss unterschieden werden, ob ein System selbst entscheiden kann, welcher Teil des Textes annotiert wird, oder ob jedes vorgegebene Objekt klassifiziert werden muss. Das einfach verkettete Zufallsfeld der MWA-Erkennung besitzt beispielsweise die Freiheit zu entscheiden, ob ein Ausdruck als MWA markiert wird, oder doch nur eine Sequenz von isolierten Wörtern darstellt. In diesem Fall können bei der Evaluation der MWA-Erkennung vier Fälle unterschieden werden.

Ergebnisse, die als **richtig positiv** (engl. *true positive*) eingestuft werden, sind Wortsequenzen, die sowohl in der Ausgabe vom System, als auch im Goldstandard als MWA markiert wurden.

Ergebnisse, die als **falsch positiv** (engl. *false positive*) eingestuft werden, sind Wortsequenzen, die von dem System als MWA markiert wurden, jedoch im Goldstandard nicht als MWA markiert wurden.

Ergebnisse, die als **falsch negativ** (engl. *false negative*) eingestuft werden, sind Wortsequenzen, die im Goldstandard als MWA markiert sind, jedoch von dem System als einzelne Wörter erkannt wurden.

Ergebnisse, die als **richtig negativ** (engl. *true negative*) eingestuft werden, sind Wörter, die von dem System als einzelne Wörter erkannt wurden und im Goldstandard ebenfalls als einzelne Wörter auftreten.

Aus diesen vier Fällen können anschließend die nachfolgenden Evaluationsmetriken berechnet werden. Die **Präzision** (engl. *precision*) beschreibt welcher Prozentsatz der Wortsequenzen, die das System als MWA markiert hat, auch wirklich als MWA im Goldstandard auftreten. Berechnet wird dies mit folgender Formel:

$$\text{Präzision} = \frac{\text{richtig Positive}}{\text{richtig Positive} + \text{falsch Positive}}$$

Die **Ausbeute** (engl. *recall*) beschreibt welcher Prozentsatz aller im Datensatz auftretenden MWAs von dem System korrekt als MWA markiert wurden. Berechnet wird dies mit folgender Formel:

$$\text{Ausbeute} = \frac{\text{richtig Positive}}{\text{richtig Positive} + \text{falsch Negative}}$$

Bei der Evaluation der Bedeutungsauflösung können in ähnlicher Weise vier Fälle unterschieden werden.

Ergebnisse, die als **richtig positiv** (engl. *true positive*) eingestuft werden, sind Wörter und MWAs, denen von UKB dieselbe Bedeutung wie im Goldstandard zugewiesen wurde.

Ergebnisse, die als **falsch positiv** (engl. *false positive*) eingestuft werden, können anhand von zwei Fällen unterschieden werden. Wörter und MWAs, bei denen die von UKB zugewiesene Bedeutung sich von der Bedeutung im Goldstandard unterscheidet, und Wörter und MWAs denen UKB eine Bedeutung zugewiesen hat, obwohl im Goldstandard keine Bedeutung vorhanden ist.

Ergebnisse, die als **falsch negativ** (engl. *false negative*) eingestuft werden, können ebenfalls anhand von zwei Fällen unterschieden werden. Wörter und MWAs, denen von UKB keine Bedeutung zugewiesen wurde, und Wörter bzw. MWAs bei denen die von UKB zugewiesene Bedeutung sich von der Bedeutung im Goldstandard unterscheidet.

Ergebnisse, die als **richtig negativ** (engl. *true negative*) eingestuft werden, sind Wörter und MWAs denen von UKB keine Bedeutung zugewiesen wurde und die auch im Goldstandard keine Bedeutung besitzen.

Aus diesen vier Fällen können anschließend Präzision und Ausbeute mit den bereits genannten Formel berechnet werden.

Sowohl bei der Evaluation der MWA-Erkennung, als auch der Bedeutungsauflösung, wird dann anhand von Präzision und Ausbeute das harmonische Mittel berechnet. Dies wird als der **F₁-Wert** bezeichnet und ist der ausschlaggebende Wert, auf den es zu achten gilt. Berechnet wird dies mit folgender Formel:

$$F_1 = \frac{2 * \text{Präzision} * \text{Ausbeute}}{\text{Präzision} + \text{Ausbeute}}$$

Betrachtet man nochmal die Definition der falsch negativen Ergebnisse bei der Bedeutungsauflösung, so ist noch eine weitere Unterscheidung möglich. In der hier genannten Definition wird auch ein falsch negatives Ergebnis erzeugt, wenn das System einem Wort, welches keine Definition in der verwendeten Wissensbasis besitzt, eine Bedeutung zuweist. Damit wird der Mangel an Informationen in einer Wissensbasis dem System als Fehler zugeschrieben. In den Namen der Tabellen wird dies als „Fehler der Wissensbasis“ bezeichnet. Es besteht jedoch auch die Möglichkeit zu sagen, dass das System eine richtige Entscheidung trifft, wenn es solchen Wörtern keine Bedeutung zuweist, da eine korrekte Bedeutung in der Wissensbasis nicht existiert. In dieser Evaluation werden beide Möglichkeiten genauer untersucht.

7.3 Evaluation der MWA-Erkennung

Für die Evaluation der MWA-Erkennung wird der in Abschnitt 7.1 beschriebene Datensatz verwendet. Hierbei besteht das Problem, dass das einfach verkettete Zufallsfeld mit diesem Datensatz trainiert wird. Eine direkte Evaluation auf diesem Datensatz ist somit nicht sinnvoll, da dies nahezu perfekte Ergebnisse erzeugt und somit nicht repräsentativ ist. Um die MWA-Erkennung dennoch auf dem kompletten Datensatz evaluieren zu können wird eine k-fache Kreuzvalidierung durchgeführt. Bei der k-fachen Kreuzvalidierung wird der Trainingsdatensatz in **k** ungefähr gleich große Teile aufgeteilt. Anschließend wird das einfach verkettete Zufallsfeld mit **k - 1** Teilen des Datensatzes trainiert, während der übrig gebliebene Datensatz für die Evaluation verwendet wird. Dies wird **k**-mal durchgeführt bis jeder Teil des Datensatzes einmal für das Testen verwendet wurde. Anschließend werden für jeden Datensatz Präzision, Ausbeute und F₁-Maß berechnet, um schließlich daraus einen Durchschnittswert zu bilden. Diese Methode besitzt den Vorteil, dass kein separater Datensatz erstellt werden muss. Für die Auswertung der Evaluationsmetriken wird die eingebaute Funktion des mwetoolkit3 verwendet. In dieser Evaluation wird eine 10-fache Kreuzvalidierung verwendet, da dies ein angemessenes Mittelmaß zwischen Aufwand und Güte der Ergebnisse darstellt.

In Tabelle 7.4 sind die Ergebnisse der 10-fachen Kreuzvalidierung des einfach verketteten Zufallsfeldes zu sehen. Betrachtet man den Durchschnitt aller Werte so wird ein Wert von 0.865 bei der Präzision, so wie ein Wert von 0.757 bei der Ausbeute erreicht. Der Durchschnittswert der Präzision ist deutlich höher als der Durchschnittswert der Ausbeute. Generell kann deshalb gesagt werden, dass die MWAs, die von dem Zufallsfeld annotiert

Tabelle 7.4: Ergebnisse der 10-fachen Kreuzvalidierung ohne Teiltreffer

Testdatensatz	Präzision	Ausbeute	F ₁ -Wert
1	0.865	0.714	0.783
2	0.841	0.791	0.815
3	0.839	0.703	0.765
4	0.801	0.602	0.688
5	0.941	0.874	0.906
6	0.845	0.839	0.842
7	0.926	0.830	0.875
8	0.885	0.777	0.827
9	0.815	0.586	0.682
10	0.887	0.855	0.871
Gesamt	0.865	0.757	0.805

werden, meist korrekt sind. Da die Ausbeute jedoch geringer ist, als die Präzision, werden viele MWAs überhaupt nicht von dem System annotiert. Dies ergibt insgesamt einen F₁-Wert von 0.805 und ist somit höher als der höchste F₁-Wert für die MWA-Erkennung bei der SemEval-2016 [SHJC16]. Dort wurde nur ein F₁-Wert von 0.611 erreicht. Mögliche Begründungen für die deutliche Steigerung des F₁-Wertes sind beispielsweise die Homogenität der MWAs innerhalb des Datensatzes. Die Anforderungsspezifikationen, mit denen der Evaluationsdatensatz erstellt wurde, sind meist kurze beschreibende Sätze. Solche Sätze bieten wenig Freiraum für komplexe Wortkonstrukte. Bei Betrachtung des Datensatzes ist erkennbar, dass vorwiegend MWAs in Form von zusammengesetzten Nomen und Eigennamen enthalten sind. Da das einfach verkettete Zufallsfeld diese Muster erlernt und bei der Annotation von MWAs einsetzt, wird insgesamt eine hohe Präzision und Ausbeute erzielt.

Generell lassen sich mehrere Fehlerquellen identifizieren, die nach der Evaluation deutlich geworden sind.

Beispiel 7.2: Beispiele für MWAs, die von dem System nicht erkannt wurden

- **LAST_BOOT_IVEC location**
- **offensive player**
- to view another ^{MWA 1}**insurance companys** ^{MWA 2}**claim data**

Beispiel 7.2 zeigt einige Beispiele für Fehler, die das System bei der MWA-Erkennung gemacht hat. Der Eigenname „LAST_BOOT_IVEC location“ wurde von dem System nicht als MWA erkannt, obwohl die meisten anderen Eigennamen erkannt wurden. Das Problem liegt hierbei vermutlich an den Unterstrichen, die nur selten in anderen Eigennamen des Datensatzes vorhanden sind, weshalb das Zufallsfeld wenig Trainingsdaten für diesen Fall besitzt. Ein weiterer Problemfall sind Adjektiv + Nomen Kombinationen wie „offensive player“, die nur selten im Datensatz auftauchen. Da das einfach verkettete Zufallsfeld vor allem mit zusammengesetzten Nomen und Eigennamen trainiert wurde, werden diese Adjektiv + Nomen Kombinationen nur selten als MWA erkannt. Zusätzlich existieren noch Folgen von MWAs wie „**insurance companys claim data**“. Das System erkennt diese zwei MWAs als einen einzelnen MWA. Dies liegt vermutlich daran, dass das Zufallsfeld durch den Datensatz darauf trainiert wurde, vor allem Nomenfolgen als MWAs zu erkennen.

Tabelle 7.5: Ergebnisse der 10-fachen Kreuzvalidierung ohne Teiltreffer und mit Suche im Wörterbuch

Testdatensatz	Präzision	Ausbeute	F ₁ -Wert
1	0.860	0.730	0.790
2	0.843	0.799	0.820
3	0.833	0.721	0.773
4	0.793	0.617	0.694
5	0.941	0.879	0.909
6	0.840	0.846	0.843
7	0.920	0.852	0.885
8	0.870	0.811	0.839
9	0.794	0.602	0.684
10	0.871	0.855	0.863
Gesamt	0.857	0.771	0.810

Wenn dann eine Nomenfolge auftaucht, die aus mehreren MWAs besteht, dann hat das System Schwierigkeiten diese zu unterscheiden.

Neben den zusammengesetzten Nomen und Eigennamen existieren in Anforderungen auch MWAs die folgende Form besitzen: *Adjektiv+ Nomen*. Diese Kombination tritt im annotierten Trainingsdatensatz im Vergleich zu den zusammengesetzten Nomen und Eigennamen selten in Form eines MWAs auf. Damit die MWA-Erkennung die wenigen Fälle dennoch korrekt annotiert, wird nun das bereits in Kapitel 5 beschriebene Verfahren angewendet. Tabelle 7.5 zeigt die Ergebnisse der 10-fachen Kreuzvalidierung des Zufallsfeldes mit zusätzlicher Suche von Adjektiv und Nomen Kombinationen. Es ist erkennbar, dass sich die Präzision des Systems bei jedem der zehn verwendeten Testdatensätze verändert hat. Dabei ist der Wert bei manchen Datensätzen gestiegen und bei manchen gefallen. Dies bedeutet, dass bei jedem Durchlauf durch die Suche im Wörterbuch noch zusätzliche MWAs annotiert wurden. Insgesamt lässt sich jedoch feststellen, dass die Präzision dadurch im Durchschnitt von 0.865 auf 0.857 gesenkt wurde. Dies liegt daran, dass einige der gefundenen Wortkombinationen zwar im Wörterbuch als MWA vorkommen, jedoch nicht im Testdatensatz als diese agieren. Nimmt man als Beispiel den Testdatensatz drei, so wurde die Wortkombination „first year“ als MWA markiert. In der Wissensquelle DBpedia wird der Begriff „first year“ auf den Eintrag über Erstsemesterstudenten umgeleitet. Im Testdatensatz drei beschreibt „first year“ jedoch lediglich ein beliebiges erstes Jahr, weshalb die Annotation inkorrekt ist. Betrachtet man hingegen die Ausbeute so erkennt man, dass diese bei allen Datensätzen, abgesehen von Datensatz 8, gestiegen ist. Dies führt zu einer Steigerung des Durchschnittswerts der Ausbeute von 0.757 auf 0.771. Insgesamt lässt sich dadurch eine Steigerung des F₁-Durchschnittswerts von 0.805 auf 0.810 beobachten.

Im Kontext des INDIRECT-Projektes ist die Nutzung der zusätzlichen MWA-Erkennung mit Hilfe des Wörterbuchs die bessere Variante. Zwar sinkt die Präzision des Systems ein wenig, jedoch ist die Steigerung der Ausbeute höher, was schließlich auch zu einem besseren F₁-Wert führt. Für ein System wie INDIRECT ist es wichtig von den vorhandenen MWAs so viele wie möglich zu erkennen, damit genügend Informationen vorhanden sind, die von dem System genutzt werden können. Es profitiert weniger von einer hohen Präzision, wenn dabei jedoch nur sehr wenige MWAs erkannt werden.

Durch die Homogenität des Trainingsdatensatzes wird ein hoher F₁-Wert bei der MWA-Erkennung erreicht. Muss das System jedoch mit Anforderungen arbeiten, die ähnliche MWAs wie in Beispiel 7.2 enthalten, so werden diese nicht korrekt erkannt. In zukünftigen

Arbeiten können die bereits genannten Fehlerquellen ausgebessert werden, indem vor allem mehr Trainingsdaten mit abwechslungsreicheren Formen von MWAs verwendet werden, damit auch diese korrekt erkannt werden können.

7.4 Evaluation der Bedeutungsauflösung

Für die Evaluation der Bedeutungsauflösung wird erneut der bereits beschriebene Datensatz verwendet. Da die Bedeutungsauflösung im Gegensatz zur MWA-Erkennung nicht auf einem maschinellen Lernverfahren basiert, das den Evaluationsdatensatz auch als Trainingsdatensatz verwendet, ist ein Verfahren wie die Kreuzvalidierung nicht nötig. Der Datensatz kann somit direkt als Referenz für die Evaluation verwendet werden.

Da bei der intrinsischen Evaluation der Bedeutungsauflösung die MWA-Erkennung noch nicht verwendet wird, müssen dem System für die Bedeutungsauflösung die zu disambiguierenden MWAs und Wörter vorgegeben werden. Für die Evaluation der DBpedia Graphen wurde hierfür über jeden Satz des Evaluationsdatensatzes iteriert und jedes Nomen, jeder Eigename und jeder markierte MWA gefiltert. In einer Kontextdatei wurde anschließend für jeden Satz ein Kontext definiert, der alle Nomen, Eigennamen und MWAs des Satzes enthält. Da der WordNet 3.1 Graph auch zwischen den Wortarten Adjektive, Verben und Adverbien unterscheidet, wurden mehrere Konfigurationen getestet, bei denen teilweise zusätzlich zu den Nomen auch alle Adjektive, Verben und Adverbien in die Kontextdatei geschrieben wurden. Die erstellten Kontextdateien wurden anschließend an das UKB-System übergeben, welches dann eine Ausgabedatei mit allen Bedeutungszuordnungen enthält.

In Abschnitt 5.7 wurden bereits einige Konfigurationsmöglichkeiten des UKB-Systems näher beschrieben. Diese werden in dieser Evaluation auf den drei Graphen, die bereits in Kapitel 6 beschrieben wurden, getestet. Des Weiteren wurde für die Evaluation der beiden Graphen mithilfe der Wörterbuchdateien jeweils eine „most frequent sense“ Grundlinie erstellt. In Kapitel 6 wurde bereits beschrieben, wie die Bedeutungen in den Wörterbuchdateien der Graphen mit Häufigkeiten annotiert wurden. Für die Grundlinie wurde jedem zu disambiguierenden Wort des Evaluationsdatensatzes, dann die Bedeutung, welche in der Wörterbuchdatei die größte Häufigkeit besitzt, zugeordnet.

Für die Ergebnisse der Evaluation ist außerdem wichtig zu erwähnen, dass von den insgesamt 1474 MWAs des Evaluationsdatensatzes, nur 269 eine vollständige Bedeutung in DBpedia besitzen. In der Wissensbasis WordNet 3.1 sind es sogar nur 120. Für 807 bzw. 1097 MWAs wurde hingegen in den Wissensbasen DBpedia bzw. WordNet 3.1 eine Bedeutung für einen Teil des MWAs gefunden, die der eigentlichen Bedeutung des MWAs nahe kommt.

In den nachfolgenden Abschnitten wird zunächst die härtere Evaluationsvariante angewendet. Dabei werden Bedeutungen, die überhaupt nicht in der Wissensquelle vorhanden sind, ebenfalls als Fehler des Systems gewertet. Ein Wort mit einem „-“ im Goldstandard erzeugt dann immer ein falsch negatives Ergebnis. Falls das System bei solchen Wörtern zusätzlich eine Ausgabe erzeugt, entsteht dazu noch ein falsch positives Ergebnis.

7.4.1 Evaluation ohne Teilbedeutungen

In diesem Abschnitt werden zunächst die Evaluationsergebnisse betrachtet, wenn Teilbedeutungen für MWAs nie akzeptiert werden. Hierfür wurden die Teilbedeutungen aus dem Goldstandard entfernt und durch ein „-“ ersetzt. Alle MWAs, die zuvor eine Teilbedeutung hatten, werden nun so gewertet, als wäre keine passende Bedeutung in der Wissensquelle vorhanden.

Tabelle 7.6: Evaluation der DBpedia Graphen (ohne Teilbedeutungen/mit Fehlern der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision	Ausbeute	F ₁ -Wert
Hypernyme + Kategorien	ppr	ja	0.228	0.211	0.219
		nein	0.295	0.272	0.283
	ppr_w2w	ja	0.403	0.373	0.387
		nein	0.374	0.346	0.359
Hypernyme	ppr	ja	0.360	0.325	0.342
		nein	0.385	0.347	0.365
	ppr_w2w	ja	0.319	0.288	0.303
		nein	0.276	0.249	0.262
Wikipedia MFS	-	-	0.292	0.267	0.279

Tabelle 7.6 zeigt die Ergebnisse der Evaluation der DBpedia Graphen, wenn die Teilbedeutungen aus dem Goldstandard entfernt wurden. Da das System generell darauf abzielt Teilbedeutungen zu finden, werden somit viele falsch positive Ergebnisse erzeugt. Den besten F₁-Wert erreicht bei dem Hypernym+Kategorien Graph die Konfiguration bei der die Berechnungsart „ppr_w2w“ und die Häufigkeiten verwendet wurden. Der erreichte Wert beträgt 0.387. Bei dem Hypernym Graph hingegen ist die Verhaltensweise umgekehrt, da die Berechnungsart „ppr“ ohne die Verwendung von Häufigkeiten mit einem F₁-Wert von 0.365 die besten Ergebnisse erzielt. Wichtig ist zu erwähnen, dass die besten Konfigurationen beider Graphen außerdem die Ergebnisse der Grundlinie in Präzision, Ausbeute und F₁-Wert übertreffen. Wie bereits in Abschnitt 5.7 näher beschrieben, wurde die Konfiguration „ppr_w2w“ eingeführt, um zu verhindern, dass zwei semantisch verwandte Bedeutungen eines Begriffs, sich gegenseitig bei der Berechnung des personalisierten PageRanks bestärken und somit den Effekt der anderen Kontextwörter vermindern. Die Konfiguration „ppr_w2w“ legt somit den Fokus stärker auf den umliegenden Kontext, anstatt auf die möglichen Bedeutungen des zu disambiguierenden Ausdrucks. Eine Vermutung weshalb der Hypernym Graph deshalb mit der Berechnungsart „ppr“ bessere Ergebnisse erzielt, ist die geringe Menge an Relationen die dieser im Vergleich zu den Hypernym + Kategorien Graph verwendet. Der Hypernym Graph verwendet 4042540 Knoten und 7795850 Kanten, während der Hypernym + Kategorien Graph 7775547 Knoten und 41087016 Kanten verwendet. Dies ist mehr als die fünffache Menge an Relationen. Wenn der Fokus der Berechnung also auf den umgebenden Kontextwörtern liegt, diese aber kaum Verbindungen zu den möglichen Bedeutungen des zu disambiguierenden Begriffs besitzen, dann kann der umgebende Kontext auch nicht präzise für die Bedeutungsauflösung genutzt werden. Da aber die Konfiguration „ppr“ weniger Gewicht auf den Kontext legt und mehr auf die möglichen Bedeutungen eines Wortes, arbeitet der kleinere Hypernym Graph besser mit dieser Konfiguration. Im Gegensatz dazu profitiert der Hypernym + Kategorien Graph vermutlich von der großen Menge an Relationen, die eine effektive Nutzung der Kontextwörter ermöglichen, wodurch bessere Ergebnisse mit der Konfiguration „ppr_w2w“ erzielt werden. Dies ist jedoch lediglich eine Vermutung basierend auf den Aussagen der Autoren Agirre und Soroa [AS09]. Zusätzlich fällt noch auf, dass die Berechnungsart „ppr“ durch die Nutzung von Häufigkeiten schlechtere Ergebnisse erzielt, wohingegen die Berechnungsart „ppr_w2w“ von den Häufigkeiten profitiert. Auch hier kann die Vermutung aufgestellt werden, dass dies an den verschiedenen Herangehensweisen beider Methoden liegt. Die Grundlinie erzielt sowohl bei der Präzision, als auch bei der Ausbeute sehr niedrige Werte, weshalb eigentlich zu erwarten ist, dass die Nutzung der Häufigkeiten bei der Auswahl einer Bedeutung einen negativen Einfluss haben. Da bei den Konfigurationen mit „ppr“,

der Kontext eine geringere Rolle spielt, liegt die Vermutung nahe, dass die Häufigkeiten die Auswahl einer passenden Bedeutung umso mehr beeinflussen, wodurch insgesamt schlechtere Ergebnisse erzielt werden. Nun stellt sich jedoch die Frage warum die Konfiguration „ppr_w2w“ dennoch von den Häufigkeiten profitiert. Eine mögliche Vermutung basiert hierbei erneut auf der Aussage der Autoren, dass die Berechnungsart „ppr_w2w“ mehr Wert auf den Kontext legt, als auf die Bedeutungen der Zielwörter selbst.

Beispiel 7.3:

Wort: person

1. Bedeutung/Häufigkeit: Grammatical_person/1151
2. Bedeutung/Häufigkeit: Person/322
3. Bedeutung/Häufigkeit: Person_(surname)/3

Beispiel 7.3 zeigt drei mögliche Bedeutungen des Wortes „person“ mit ihren absoluten Häufigkeiten. Während die Konfigurationen, welche die Häufigkeiten in Kombination mit der Berechnungsart „ppr“ verwenden, in den meisten Fällen die inkorrekte Bedeutung „Grammatical_person“ zuweisen, wird bei der Konfiguration mit der Berechnungsart „ppr_w2w“ trotz Verwendung von Gewichten die korrekte Bedeutung „Person“ zugewiesen. Die Vermutung liegt deshalb nahe, dass bei der Berechnungsart „ppr_w2w“ schon aufgrund des umgebenden Kontextes die Bedeutung „Grammatical_person“ ausgeschlossen wird und die Gewichte dann lediglich dabei helfen eine sehr unwahrscheinliche Bedeutung wie „Person_(surname)“ auszuschließen. Dadurch können die Gewichte dann bei der Berechnungsart „ppr_w2w“ effektiv genutzt werden.

Des Weiteren ist deutlich zu erkennen, dass die Ausbeute aller Konfigurationen sehr niedrig ist. Dies liegt vor allem daran, dass sowohl alle MWAs mit Teilbedeutungen, als auch MWAs ohne Bedeutung in der Wissensbasis, falsch negative Ergebnisse erzeugen. Da nur 289 der 1474 MWAs eine vollständige Bedeutung in DBpedia besitzen, wirkt sich dies stark negativ auf die Ausbeute aus.

Betrachtet man einige Fehlerquellen, so wird die bereits beschriebene ineffektive Nutzung von Häufigkeiten bei der Berechnungsart „ppr“ deutlich. Ein weiteres Beispiel hierfür ist das Wort „error“, welches im Goldstandard die Bedeutung „Error“ trägt, und bei der Berechnungsart „ppr“ die Bedeutung „Error_(baseball)“ erhält. Dies liegt vor allem daran, dass die Häufigkeit von „Error_(baseball)“ 1332 beträgt, während sie bei der korrekten Bedeutung „Error“ nur 208 beträgt. Bei der Berechnungsart „ppr_w2w“ hingegen wird bei beiden Graphen selbst mit der Nutzung von Gewichten, die korrekte Bedeutung zugewiesen. Des Weiteren gibt es Fälle wie den MWA „error code sequence“, der nicht vollständig in der Wissensbasis DBpedia vorhanden ist. Stattdessen wurde im Goldstandard die angemessene Teilbedeutung „Error_code“ zugewiesen. Da das System jedoch immer ausgehend vom Kopf des Ausdrucks nach einer passenden Teilbedeutung sucht, wird die Bedeutung „Sequence“ zugewiesen, welche in diesem Fall nicht die wahre Bedeutung des Ausdrucks widerspiegelt. Zukünftige Arbeiten können also die Suche nach Teilbedeutungen verbessern, indem sie berücksichtigen, dass der Kopf des Ausdrucks nicht immer bei der Suche nach einer angemessenen Bedeutung involviert sein muss. Außerdem können zukünftige Arbeiten versuchen eine bessere Gewichtung der Bedeutungen zu finden, damit diese effektiv genutzt werden können.

In Tabelle 7.7 sind die Evaluationsergebnisse für den WordNet 3.1 Graph zu sehen, wenn Teilbedeutungen nicht zugelassen werden. Im Gegensatz zu der Wissensbasis DBpedia wur-

Tabelle 7.7: Evaluation des WordNet 3.1 Graphen (ohne Teilbedeutungen/mit Fehlern der Wissensbasis)

Kontext	PageRank	Häufigkeit	Präzision	Ausbeute	F ₁ -Wert
Adj, Adv, V, N, MWA	ppr	ja	0.454	0.414	0.433
		nein	0.304	0.277	0.290
	ppr_w2w	ja	0.418	0.382	0.399
		nein	0.345	0.315	0.329
N, MWA	ppr	ja	0.450	0.411	0.430
		nein	0.302	0.275	0.288
	ppr_w2w	ja	0.421	0.384	0.402
		nein	0.352	0.321	0.336
Adj, Adv, V, N, MWA (N)	ppr	ja	0.465	0.421	0.442
		nein	0.306	0.278	0.291
	ppr_w2w	ja	0.426	0.386	0.405
		nein	0.354	0.321	0.337
N, MWA (N)	ppr	ja	0.461	0.418	0.438
		nein	0.304	0.275	0.289
	ppr_w2w	ja	0.426	0.386	0.405
		nein	0.359	0.325	0.341
WordNet 3.1 MFS	-	-	0.406	0.372	0.388

den hier nicht verschiedene Graphen evaluiert, sondern nur verschiedene Möglichkeiten den Kontext für die Bedeutungsauflösung zu wählen. Die beste Konfiguration verwendet dabei Adjektive (Adj.), Adverbien (Adv.), Verben (V), Nomen (N) und MWAs für den Kontext und erreicht einen F₁-Wert von 0.442. Wichtig ist zu erwähnen, dass bei den Konfigurationen mit dem Bezeichner „MWA (N)“, alle MWAs als Nomen markiert wurden. Ähnlich wie bei dem Hypernym DBpedia Graph erzielt die Konfiguration „ppr“ deutlich bessere Ergebnisse, als die Konfiguration „ppr_w2w“. Es kann erneut die Vermutung aufgestellt werden, dass dies an der geringen Menge an Relationen innerhalb des Graphen liegt. Die Autoren Agirre und Soroa [AS09] verwenden in ihrer Evaluation beispielsweise den WordNet 3.0 Graphen mit zusätzlichen „gloss relations“. Da in dieser Arbeit jedoch der WordNet 3.1 als Wissensbasis verwendet wird, sind keine „gloss relations“ vorhanden, die für den Graphen verwendet werden könnten. Somit ergibt sich, dass der hier verwendete Graph 116787 Knoten und nur 184083 Kanten verwendet. Der WordNet 3.0 Graph der Autoren verwendet hingegen 117522 Knoten und dafür aber 525356 Kanten [AS09]. Im Gegensatz zu den DBpedia Graphen erzielen jedoch alle Konfigurationen bessere Ergebnisse, wenn die Häufigkeiten für die Bedeutungsauflösung verwendet werden. Wie zu erkennen ist, sind die Ergebnisse der Grundlinie des WordNet 3.1 Graphen deutlich besser, als die Ergebnisse der Grundlinie der DBpedia Graphen. Somit kann die Vermutung aufgestellt werden, dass die verwendeten Häufigkeiten generell einen positiven Einfluss auf die Bedeutungsauflösung haben, weshalb auch alle Konfigurationen davon profitieren. Des Weiteren ist auch hier zu sehen, dass die Ausbeute in allen Konfigurationen niedrigere Werte, als die Präzision erreicht. Dies lässt sich hier auch hier anhand der großen Menge an MWAs, die entweder nur eine Teilbedeutung, oder keine Bedeutung im Goldstandard besitzen. Da für den WordNet 3.1 Graph sogar nur für 120 der 1474 MWAs eine vollständige Bedeutung im Goldstandard annotiert wurde, steigt die Menge an falsch negativen Ergebnissen noch stärker als bei den DBpedia Graphen.

7.4.2 Evaluation mit Teilbedeutungen

Wie bereits in Abschnitt 5.6.2 näher analysiert wurde, entsteht ein Informationsverlust, wenn Teilbedeutungen eines MWAs als inkorrekt angesehen werden. Es ist deshalb sinnvoll auch Teilbedeutungen für die MWAs, als korrekt zu werten, wenn keine vollständige Bedeutung in der Wissensquelle vorhanden ist. Die Teilbedeutungen können nämlich durch INDIRECT genutzt werden, um zumindest einen Teil der Informationen, die in MWAs enthalten sind, zu nutzen. Da das implementierte WSD-System vor allem darauf konzipiert wurde auch solche Teilbedeutungen zu finden, spiegeln die nachfolgenden Ergebnisse eher die Performanz des Systems wider.

In Tabelle 7.8 sind die Evaluationsergebnisse der beiden verwendeten DBpedia Graphen zu sehen, wenn die Teilbedeutungen im Goldstandard zugelassen werden. Dabei ist zu beachten, dass bei der Evaluation beider Graphen, die Größe des Kontextfensters variiert wurde. Die linke Seite der Spalten Präzision, Ausbeute und F_1 -Wert enthalten die Ergebnisse, bei dem für den Kontext nur ein Satz verwendet wurde. Die rechte Seite hingegen zeigt die Ergebnisse, wenn für den Kontext drei Sätze verwendet wurden. Es ist zunächst erkennbar, dass alle Konfigurationen bessere Ergebnisse bei Präzision, Ausbeute und F_1 -Wert erzielen, als in Tabelle 7.6. Dies liegt daran, dass die Teilbedeutungen im Goldstandard nun zugelassen sind und somit die Teilbedeutungen des Systems nur dann als falsch positiv gewertet werden, wenn sie auch wirklich inkorrekt sind. Auch hier wiederum erzielen fast alle Konfigurationen der beiden Graphen einen höheren F_1 -Wert, als die „most frequent sense“ Grundlinie. Der Graph „Hypernyme + Kategorien“ mit der Berechnungsart „ppr_w2w“ und der Nutzung von Worthäufigkeiten erzielt mit einem F_1 -Wert von 0.496 bei einer Kontextgröße von einem einzigen Satz und einem F_1 -Wert von 0.506 bei einer Kontextgröße von drei Sätzen, die besten Ergebnisse. Obwohl sich die Ergebnisse bei Präzision, Ausbeute und F_1 -Wert von den Ergebnissen in Tabelle 7.6 unterscheiden, ändert sich das Verhalten der Graphen im Bezug auf die Berechnungsarten und die Verwendung von Häufigkeiten nicht. Vergleicht man die Ergebnisse der Graphen bei dem ein einziger Satz als Kontext dient, mit den Ergebnissen, bei denen für den Kontext drei Sätze verwendet wurden, so ist nur eine leichte Steigerung der F_1 -Werte zu erkennen. Diese Steigerung lässt jedoch die Vermutung zu, dass ein noch größeres Kontextfenster zu weiteren Verbesserungen führt. In zukünftigen Arbeiten ist dies deshalb ein Ansatzpunkt für weitere mögliche Leistungssteigerungen.

Tabelle 7.8: Evaluation der DBpedia Graphen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision		Ausbeute		F_1 -Wert	
Hypernyme + Kategorien	ppr	ja	0.298	0.299	0.275	0.276	0.286	0.287
		nein	0.383	0.392	0.354	0.362	0.368	0.376
	ppr_w2w	ja	0.516	0.527	0.477	0.486	0.496	0.506
		nein	0.478	0.485	0.442	0.448	0.459	0.466
Hypernyme	ppr	ja	0.452	0.454	0.407	0.409	0.428	0.430
		nein	0.483	0.488	0.435	0.439	0.458	0.462
	ppr_w2w	ja	0.411	0.421	0.371	0.379	0.390	0.399
		nein	0.359	0.383	0.323	0.345	0.340	0.363
Wikipedia MFS	-	-	0.360	-	0.329	-	0.344	-
Babelify (S)	-	-	0.517	-	0.292	-	0.373	-
Babelify	-	-	0.476	-	0.292	-	0.362	-

Um die Ergebnisse des UKB-Systems besser einschätzen zu können wurde noch eine weitere Evaluation mit dem wissensbasierten WSD-System Babelfy durchgeführt. Babelfy annotiert einen Eingabetext sowohl mit Bedeutungen aus der Wissensquelle Wordnet 3.0, als auch mit DBpedia URLs. Babelfy erstellt dabei semantische Annotationen, die mehrere Wörter überspannen können. Es besteht außerdem die Möglichkeit, dass für eine Spanne von Wörtern mehrere Bedeutungen annotiert werden. Um einen fairen Vergleich mit dem UKB-System zu ermöglichen, wurde bei allen Evaluationen mit Babelfy geprüft, ob mindestens eine der vergebenen Annotationen mit dem Goldstandard übereinstimmt. Falls dies der Fall ist, wurde dies als richtig positives Ergebnis gewertet. Bei der Variante, die mit einem (S) markiert ist, wird außerdem jede zusätzliche Annotation nicht als falsch positives Ergebnis gewertet. Bei der Variante ohne die Markierung (S) wird jede zusätzliche Annotation als falsch positive gewertet. Betrachtet man die Variante, bei der keine zusätzlichen falsch positiven Ergebnisse gewertet werden, so wird deutlich, dass Babelfy mit einer Präzision von 0.517 sogar besser ist, als die beste Präzision des UKB-Systems. Dabei werden aber alle zusätzlichen Annotationen ignoriert. Bezieht man diese in die Evaluation mit ein, so sinkt die Präzision auf 0.476. Des Weiteren ist zu beachten, dass eine hohe Präzision für ein WSD-System alleine nicht ausreichend ist. Wenn ein WSD-System zwar bei den annotierten Bedeutungen eine hohe Präzision erreicht, jedoch insgesamt wenig Bedeutungen annotiert, so gewinnt man dennoch wenig Informationen. Eine hohe Ausbeute ist deshalb, vor allem auch im Rahmen des INDIRECT-Projektes, sehr wichtig. Babelfy erreicht dabei jedoch nur einen Wert von 0.292. Dies ist schlechter als sieben der acht Konfigurationen der DBpedia Graphen, die das UKB-System verwendet. Die niedrige Ausbeute entsteht durch die geringe Menge von Annotationen mit DBpedia URLs, die von dem Babelfy System ausgegeben werden. Insgesamt erreicht Babelfy nur einen F_1 -Wert von 0.373 und ist somit deutlich schlechter als der beste F_1 -Wert des WSD-Systems dieser Arbeit mit 0.496.

Insgesamt lässt sich sagen das der beste F_1 -Wert 0.496, der durch UKB mit dem Hypernyme + Kategorien Graph erreicht wird, bereits angemessen ist für die Nutzung im Rahmen des INDIRECT Projektes. Dabei ist jedoch zu beachten, dass die Präzision auch hier in allen Fällen höher ist, als die Ausbeute. Eine hohe Präzision gibt zwar an, wie viele der annotierten Bedeutungen korrekt sind, jedoch ist daran nicht zu erkennen, wie viele der benötigten Bedeutungen auch wirklich annotiert wurden. Da das INDIRECT-Projekt eine große Menge an Informationen aus den Anforderungen gewinnen will, ist deshalb eine hohe Ausbeute nötig. Die Ausbeute leidet jedoch in dieser Evaluation vor allem an den fehlenden Definitionen der Wissensbasis. Da insgesamt 864 der Wörter und Ausdrücke keine Bedeutung im Goldstandard haben, werden diese automatisch als falsch negative Ergebnisse gewertet, wodurch die Ausbeute stark sinkt. Dennoch ist der beste Wert der Ausbeute mit 0.477 akzeptabel. Auch im Vergleich zu einem anderen state of the art WSD-System wie Babelfy, welches ebenfalls DBpedia als Wissensbasis verwendet, werden deutlich bessere Ergebnisse erzeugt. Des Weiteren kann man feststellen, dass UKB bereits in anderen Arbeiten [ALS18] mit der Wissensquelle WordNet einen maximalen F_1 -Wert von 0.703 erreicht hat. Dies zeigt, das womöglich noch einige Verbesserungsmöglichkeiten für zukünftige Arbeiten existieren.

In Tabelle 7.9 sind die Evaluationsergebnisse des WordNet 3.1 Graphen zu sehen, wenn Teilbedeutungen im Goldstandard zugelassen werden. Wie schon zuvor erklärt enthält auch diese Tabelle sowohl die Ergebnisse für den Kontext, der nur aus einem Satz besteht, als auch die Ergebnisse für ein Kontextfenster mit drei Sätzen. Die Anordnung ist dieselbe wie in Tabelle 7.8. Zunächst einmal ist auch hier eine deutliche Steigerung des Präzision, Ausbeute und somit auch des F_1 -Wertes im Vergleich zu Tabelle 7.7 zu erkennen. Dies liegt auch hier an der geringeren Menge an falsch positiven Ergebnissen und falsch negativen Ergebnissen, die durch die Teilbedeutungen entstanden sind. Auch hier ändert sich

Tabelle 7.9: Evaluation des WordNet 3.1 Graphen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)

Kontext	PageRank	Häufigkeit	Präzision		Ausbeute		F ₁ -Wert	
Adj., Adv., V, N, MWA	ppr	ja	0.559	0.558	0.510	0.508	0.533	0.532
		nein	0.373	0.378	0.340	0.345	0.356	0.361
	ppr_w2w	ja	0.509	0.510	0.464	0.465	0.485	0.486
		nein	0.411	0.402	0.375	0.366	0.392	0.383
N, MWA	ppr	ja	0.556	0.554	0.507	0.505	0.530	0.528
		nein	0.371	0.379	0.338	0.345	0.354	0.361
	ppr_w2w	ja	0.509	0.522	0.464	0.476	0.485	0.498
		nein	0.420	0.418	0.383	0.382	0.401	0.399
Adj., Adv., V, N, MWA (N)	ppr	ja	0.575	0.573	0.521	0.519	0.547	0.545
		nein	0.379	0.385	0.344	0.349	0.361	0.366
	ppr_w2w	ja	0.525	0.523	0.475	0.474	0.499	0.497
		nein	0.429	0.422	0.389	0.382	0.408	0.401
N, MWA (N)	ppr	ja	0.571	0.568	0.517	0.514	0.543	0.540
		nein	0.375	0.384	0.340	0.348	0.357	0.365
	ppr_w2w	ja	0.523	0.537	0.474	0.487	0.497	0.511
		nein	0.436	0.437	0.395	0.396	0.414	0.415
WordNet 3.1 MFS	-	-	0.504	-	0.460	-	0.481	-

das Verhalten des Graphen im Bezug auf die Berechnungsarten und die Verwendung von Häufigkeiten nicht. Das beste Ergebnis, mit einem F₁-Wert von 0.547, erreicht die Konfiguration bei der Adjektive, Adverbien, Verben, Nomen und MWAs als Kontext genutzt und die Berechnungsart „ppr“ in Kombination mit den Häufigkeiten verwendet werden. Dies ist besser, als das beste Ergebnis der DBpedia Graphen. Insgesamt lässt sich sagen, dass auch der hier erreichte maximale F₁-Wert von 0.547 für die Nutzung im Rahmen des INDIRECT-Projektes angemessen ist.

Um auch hier einen Vergleich zu Babelfy zu erfassen, wurde auch eine Evaluation mit den WordNet-Synsets, die durch Babelfy ausgegeben werden, durchgeführt. Durch die täglichen Begrenzungen der Babelnet Anwendung war es jedoch nur möglich die ersten 100 Anforderungen des Evaluationsdatensatzes zu nutzen. Da Babelnet außerdem nur Synsets für WordNet 3.0 oder niedriger ausgibt, wurde eine automatische Abbildung zwischen beiden Versionen verwendet [Zen19]. Es kann also nicht in allen Fällen garantiert werden, dass die Abbildungen korrekt sind.

Tabelle 7.10 zeigt den Vergleich der Ergebnisse von Babelfy und dem WordNet 3.1 auf Graphen bei den ersten 100 Anforderungen des Evaluationsdatensatzes. Auch hier gilt, dass bei der Evaluationsvariante, die mit einem (S) markiert ist, alle zusätzlichen Annotationen nicht als falsch positives Ergebnis gewertet werden. Für den Vergleich wurde nur die beste Konfiguration des WordNet 3.1 Graphen aus Tabelle 7.9 verwendet. Wie zu erkennen ist erreicht die beste Babelfy Variante einen F₁-Wert von 0.525 und ist somit besser als die getestete Version des WordNet 3.1 Graphen. Bezieht man jedoch alle zusätzliche Annotationen von Babelfy in die Evaluation mit ein, so sinkt dieser Wert auf 0.462. Der WordNet 3.1 Graph erzielt in diesem Fall mit einem maximalen F₁-Wert von 0.485 die besseren Ergebnisse. Insgesamt ist feststellbar, dass bei der Evaluation mit der Wissensbasis WordNet, die Ergebnisse beider Systeme deutlich näher beieinander liegen. Dies liegt vor allem daran, dass Babelfy durch eine deutlich größere Menge an Annotationen eine bessere Ausbeute erzielt, als mit der Wissensquelle DBpedia.

Tabelle 7.10: Evaluation des WordNet 3.1 Graphen mit den ersten 100 Anforderungen (mit Teilbedeutungen/mit Fehlern der Wissensbasis)

Kontext	PageRank	Häufigkeit	Präzision	Ausbeute	F ₁ -Wert
Adj.,	ppr	ja	0.544	0.437	0.485
Adv., V,		nein	0.437	0.351	0.389
N, MWA	ppr_w2w	ja	0.482	0.387	0.429
(N)		nein	0.386	0.310	0.344
Wikipedia MFS	-	-	0.473	0.382	0.423
Babelify (S)	-	-	0.601	0.466	0.525
Babelify	-	-	0.459	0.466	0.462

7.4.3 Evaluation ohne Fehler der Wissensbasis

Im nachfolgenden Abschnitt wird eine Evaluation durchgeführt bei der fehlende Definitionen der Wissensquelle nicht immer sofort als Fehler des Systems angesehen werden. Wie zuvor bei der härteren Evaluation deutlich geworden ist, ist die Ausbeute des Systems meist niedriger, als die Präzision. Dies liegt vor allem daran, dass 864 bzw. 517 der Wörter und MWAs in den Wissensbasen DBpedia bzw. WordNet 3.1 keine passende Bedeutung besitzen. Für das System existiert in diesem Fall nicht die Möglichkeit eine korrekte Ausgabe zu erzeugen. In der folgenden Evaluation, werden fehlende Definition in den Wissensbasen nicht immer als Fehler des Systems angesehen. Wenn das System bei einer fehlenden Definition auch keine Ausgabe erzeugt, so wird dies als richtig negatives Ergebnis angesehen. Wird bei einem solchen Wort eine falsche Bedeutung ausgegeben, so wird dies lediglich als falsch positives und nicht als falsch negatives Ergebnis gewertet. Diese Evaluation ist ein Einblick darin wie gut das System unabhängig von den Wissensbasen Bedeutungen auflöst. Wichtig ist hierbei zu beachten, dass die Präzision sich im Vergleich zur härteren Evaluation nicht verändert, da jede inkorrekte Ausgabe auch hier als inkorrekt gewertet wird.

Tabelle 7.11 zeigt die Ergebnisse der Evaluation der DBpedia Graphen, wenn Informationslücken der Wissensbasis DBpedia nicht als Fehler des Systems gewertet werden. Der

Tabelle 7.11: Evaluation der DBpedia Graphen (mit Teilbedeutungen/ohne Fehler der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision		Ausbeute		F ₁ -Wert	
Hypernyme	ppr	ja	0.298	0.299	0.351	0.352	0.322	0.323
		nein	0.383	0.392	0.451	0.462	0.414	0.424
+ Kategorien	ppr_w2w	ja	0.516	0.527	0.608	0.620	0.558	0.570
		nein	0.478	0.485	0.564	0.572	0.517	0.525
Hypernyme	ppr	ja	0.452	0.454	0.519	0.522	0.483	0.486
		nein	0.483	0.488	0.555	0.561	0.517	0.522
	ppr_w2w	ja	0.411	0.421	0.473	0.484	0.440	0.450
		nein	0.359	0.383	0.412	0.440	0.384	0.410
Wikipedia MFS	-	-	0.360	-	0.419	-	0.387	-
Babelify (S)	-	-	0.517	-	0.372	-	0.433	-
Babelify	-	-	0.476	-	0.372	-	0.418	-

Aufbau der Tabelle ist äquivalent zu dem Aufbau von Tabelle 7.8. Wie zu erkennen ist, verändert sich die Präzision des Systems nicht, da jede fehlerhafte Ausgabe auch hier als falsch positives Ergebnis gewertet wird. Betrachtet man jedoch die Ausbeute, so ist eine deutliche Verbesserung in allen Fällen zu erkennen. Dies liegt vor allem daran, dass für die 864 Wörter und Ausdrücke, die keine passende Bedeutung in DBpedia besitzen, nun auch keine Ausgabe erwartet wird. Den maximalen Wert bei der Ausbeute erreicht auch hier wieder der Hypernyme + Kategorien Graph mit einem Wert von 0.608 bzw. 0.620. Dies ist eine deutliche Steigerung gegenüber den Werten 0.477 bzw. 0.486, die zuvor erreicht wurden. Was die Tabelle zeigen soll, ist das die Ausbeute des Systems deutlich durch die Wahl der Wissensbasis eingeschränkt ist. DBpedia eignet sich primär für Fachbegriffe wie „database management system“ oder „denial of service attack“. Bei einzelnen Begriffen wie „type“, „ability“ oder auch „status“ fehlen Definitionen, die sich auf einen allgemeinen Kontext anwenden lassen. Beispielsweise gibt es für den Begriff „status“ Definitionen wie „Legal_status“, „Martial_status“ oder auch „Conservation_status“. Diese sind jedoch sehr spezifisch und sind nicht anwendbar auf allgemeine Kontexte in denen einfach der Zustand einer Sache beschrieben wird.

Tabelle 7.12 zeigt die Evaluation des WordNet 3.1 Graphen, wenn Informationslücken der Wissensbasis WordNet nicht als Fehler des Systems gewertet werden. Auch hier ist der Aufbau äquivalent zu Tabelle 7.9 und die Präzision hat sich in keiner Konfiguration verändert. Da in der WordNet 3.1 Wissensbasis nur 517 der Wörter und MWAs keine passende Bedeutung besitzen, ist die Ausbeute im Vergleich zu den DBpedia Graphen nicht so stark gestiegen. Hier erreicht die beste Konfiguration beispielsweise eine Ausbeute von 0.598 bzw. 0.596. Im Vergleich dazu erreichte sie in Tabelle 7.9 eine Ausbeute von 0.521 bzw. 0.519. Obwohl dennoch eine deutliche Verbesserung zu sehen ist, zeigt dies, dass die Wissensbasis WordNet 3.1 das System insgesamt bei der Bedeutungsauflösung weniger einschränkt. Die zuvor genannten Begriffe wie „type“, „ability“ und „status“ besitzen alle eine passende Bedeutung für einen allgemeinen Kontext in WordNet.

Tabelle 7.12: Evaluation des WordNet 3.1 Graphen (mit Teilbedeutungen/ohne Fehler der Wissensbasis)

Kontext	PageRank	Häufigkeit	Präzision		Ausbeute		F ₁ -Wert	
Adj., Adv., V, N, MWA	ppr	ja	0.559	0.558	0.585	0.583	0.572	0.570
		nein	0.373	0.378	0.390	0.395	0.381	0.386
	ppr_w2w	ja	0.509	0.510	0.533	0.534	0.521	0.522
		nein	0.411	0.402	0.430	0.420	0.420	0.411
N, MWA	ppr	ja	0.556	0.554	0.581	0.579	0.568	0.566
		nein	0.371	0.379	0.388	0.396	0.379	0.387
	ppr_w2w	ja	0.509	0.522	0.533	0.546	0.521	0.534
		nein	0.420	0.418	0.439	0.438	0.429	0.428
Adj., Adv., V, N, MWA (N)	ppr	ja	0.575	0.573	0.598	0.596	0.586	0.584
		nein	0.379	0.385	0.394	0.401	0.386	0.393
	ppr_w2w	ja	0.525	0.523	0.545	0.543	0.535	0.533
		nein	0.429	0.422	0.447	0.438	0.438	0.430
N, MWA (N)	ppr	ja	0.571	0.568	0.593	0.590	0.582	0.579
		nein	0.375	0.384	0.390	0.399	0.382	0.391
	ppr_w2w	ja	0.523	0.537	0.544	0.559	0.533	0.548
		nein	0.436	0.437	0.453	0.454	0.444	0.445
WordNet 3.1 MFS	-	-	0.504	-	0.528	-	0.516	-

Insgesamt lässt sich also feststellen, dass sich die Informationslücken beider Wissensbasen negativ auf die Ausbeute der Bedeutungsauflösung auswirken. In zukünftigen Arbeiten können also Verbesserungen angestrebt werden, indem die Wissensbasen erweitert, ausgetauscht, oder in Kombination miteinander verwendet werden, damit das System zumindest die Chance besitzt, überhaupt eine korrekte Bedeutung zuweisen zu können.

7.5 Kombinierte Evaluation

Nachdem sowohl eine intrinsische Evaluation der MWA-Erkennung, als auch der Bedeutungsauflösung durchgeführt wurde, wird nun eine kombinierte Evaluation durchgeführt bei der das Zusammenspiel beider Teilsysteme getestet wird. Die intrinsischen Evaluationen helfen dabei Fehlerquellen beider Teilsysteme zu erkennen und mögliche Verbesserungsvorschläge zu nennen. Da die Bedeutungsauflösung jedoch im Rahmen des INDIRECT-Projektes nicht isoliert von der MWA-Erkennung arbeitet, ist sie immer abhängig von dessen Ergebnissen. Nur eine kombinierte Evaluation gibt also den nötigen Einblick, um konkrete Schlüsse bezüglich der wahren Performanz des Systems zu ziehen. Da auch hier erneut der bereits beschriebene Datensatz verwendet und dieser, wie bereits beschrieben, auch als Trainingsdatensatz für das einfach verkettete Zufallsfeld der MWA-Erkennung verwendet wurde, kann keine direkte Evaluation auf dem Datensatz stattfinden. Stattdessen werden die zehn annotierten Datensätze der Kreuzvalidierung aus Abschnitt 7.3 verwendet, um die Eingabe für die Bedeutungsauflösung zu erzeugen. Für jeden Satz aus den zehn Datensätzen werden hierfür die markierten MWAs, die Nomen und die Eigennamen extrahiert und es wird ähnlich wie bei der Evaluation der Bedeutungsauflösung eine Kontextdatei erzeugt, die anschließend von UKB verarbeitet wird. Für die kombinierte Evaluation werden dieselben Graphen und deren Konfigurationen, wie bereits in Abschnitt 7.4, verwendet.

Um hier eine Unterscheidung zwischen verschiedenen Fällen durchzuführen, ist es zunächst wichtig zu verstehen, worauf in dieser Evaluation Wert gelegt wird. Obwohl die MWA-Erkennung ein wichtiger Bestandteil dieser Arbeit ist, sind es eigentlich die Informationen der Bedeutungsauflösung, die vor allem für INDIRECT relevant sind. Bei der Bewertung werden deshalb neben Annotationen bei denen sowohl die MWA-Markierung, als auch die Bedeutung korrekt sind, auch Annotationen bei denen der MWA falsch markiert, aber dennoch die Bedeutung korrekt annotiert wurden, als richtig positive Ergebnisse gewertet. Nimmt man als Beispiel den MWA „error code sequence“, so ist die korrekte Bedeutung in der Wissensbasis DBpedia „Error_code“. Falls das System diese Bedeutung annotiert, jedoch als MWA nur den Teil „error code“ markiert hat, so wird dies dennoch als richtig positives Ergebnis gewertet, da die Bedeutung korrekt ist. Wird jedoch noch zusätzlich für das Wort „sequence“ eine Bedeutung annotiert, so wird diese Annotation als zusätzliches falsch positives Ergebnis gewertet. Die anderen Fälle sind äquivalent zu den bereits verwendeten Fällen bei der Bedeutungsauflösung.

Die nachfolgenden Ergebnisse stellen eine Evaluation mit Teilbedeutungen und mit den Fehlern, die durch die Wissensbasis entstehen dar. Diese Evaluationsweise wurde gewählt, da sie die Performanz des Systems am präzisesten darstellt. Um dennoch die Ergebnisse der anderen Evaluationsvarianten zu sehen, wurden die Tabellen Tabelle A.1 und Tabelle A.1 für die DBpedia Graphen, so wie die Tabellen Tabelle A.3 und Tabelle A.3 für den WordNet 3.1 Graphen im Anhang hinzugefügt.

In Tabelle 7.13 sind die Evaluationsergebnisse der kombinierten Evaluation mit DBpedia Graphen zu sehen. Insgesamt ist erkennbar, dass bei allen Konfigurationen sowohl die Präzision als auch die Ausbeute in einem Bereich zwischen 2% und 3% gesunken ist. Bei der Grundlinie ist sogar eine Verschlechterung zwischen 3% und 4% zu sehen. Vor allem bei der Präzision war dies zu erwarten, da einige neue falsch positive Ergebnisse durch die inkorrekt

Tabelle 7.13: Vergleich der kombinierten Ergebnisse (DBpedia/mit Teilebedeutungen/mit Fehlern der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision	Ausbeute	F ₁ -Wert
Hypernyme + Kategorien	ppr	ja	0.278	0.258	0.268
		nein	0.359	0.334	0.346
	ppr_w2w	ja	0.488	0.453	0.470
		nein	0.450	0.418	0.433
Hypernyme	ppr	ja	0.425	0.385	0.404
		nein	0.456	0.413	0.433
	ppr_w2w	ja	0.382	0.346	0.363
		nein	0.334	0.303	0.318
Wikipedia MFS	-	-	0.324	0.298	0.310

markierten MWAs entstehen. Betrachtet man beispielsweise den Ausdruck „Ping-Pong Timeout interrupt“, so existiert dafür kein vollständiger Eintrag in den Wissensquellen. Die eigentliche Bedeutung des Ausdrucks wird von dem Kopf „interrupt“ getragen, weshalb das System nach dieser Teilbedeutung suchen und diese dem gesamten Ausdruck zuweisen sollte. Da das System jedoch das Wort „Ping-Pong“ von dem Ausdruck „Timeout interrupt“ trennt, wird beiden eine Bedeutung zugewiesen. Der Ausdruck „Timeout interrupt“ erhält die korrekte Bedeutung „Interrupt“. Dem Wort „Ping-Pong“ wird jedoch die Bedeutung „Table_tennis“ zugewiesen, was dann als falsch positives Ergebnis gewertet wird. Neben den vielen neuen falsch positiven Ergebnissen sind auch einige falsch negative Ergebnisse durch die falsch annotierten MWAs hinzugekommen. Nimmt man beispielsweise den MWA „critical_path“, so muss diesem die Bedeutung „Critical_path_method“ zugewiesen werden. Da dieser MWA jedoch von dem System nicht erkannt wird, erhält lediglich das Wort „path“ die Bedeutung „Footpath“. Dies erzeugt insgesamt ein falsch positives und ein falsch negatives Ergebnis. Eine solche Verschlechterung der beiden Werte war zu erwarten, da die MWA-Erkennung bei der Kreuzvalidierung im Durchschnitt nur einen F₁-Wert von 0.810 erreicht hat. Da sich alle Konfigurationen relativ konstant verschlechtert haben, erreicht auch hier der Hypernyme + Kategorien Graph mit der Berechnungsart „ppr_w2w“ und der Verwendung von Häufigkeiten den höchsten F₁-Wert mit 0.470.

Insgesamt lässt sich sagen, dass nur eine geringe Verschlechterung der Ergebnisse nach der Kombination beider Teilsysteme erkennbar ist. Mit einem maximalen F₁-Wert mit 0.529 erzielt das gesamte System akzeptable Ergebnisse für die Nutzung im Rahmen des INDIRECT Projektes. Die hier entstandenen Fehler sind eine Kombination der bereits in den intrinsischen Evaluation beschriebenen Fehlerquellen. Die Präzision sinkt stark, durch die Menge an falsch Positiven Ergebnissen, die durch die inkorrekte Annotation von MWAs entsteht. Des Weiteren ist Verbesserungspotential bei der Auswahl einer passenden Teilbedeutung vorhanden. Hierfür müsste untersucht werden, ob neben dem Kopf noch weitere Informationsträger innerhalb eines Ausdrucks vorhanden sind, die für die Auswahl einer passenden Teilbedeutung verwendet werden könnten.

Tabelle 7.14 zeigt die Evaluationsergebnisse der kombinierten Evaluation mit dem Word-Net Graphen. Auch hier ist eine ähnliche Verschlechterung der Präzision und Ausbeute in einem Bereich zwischen 2% und 3% zu erkennen. Die Ergebnisse der Grundlinie verschlechtern sich erneut in einem Bereich zwischen 3% und 4%. Mit einem F₁-Wert von 0.523 erzielt die Konfiguration, die Adjektive, Verben, Adverbien und Nomen des Kontextes verwendet, auch hier die besten Ergebnisse. Bei der Verwendung der Wissensbasis treten ähnliche Fehler auf, wie sie auch bei der Evaluation der DBpedia Graphen beschrieben

Tabelle 7.14: Vergleich der kombinierten Ergebnisse (WordNet/mit Teilbedeutungen/mit Fehlern der Wissensbasis)

Kontextwortarten	PageRank	Häufigkeit	Präzision	Ausbeute	F ₁ -Wert
Adj., Adv., V, N, MWA	ppr	ja	0.536	0.489	0.511
		nein	0.358	0.327	0.342
	ppr_w2w	ja	0.488	0.446	0.466
		nein	0.400	0.365	0.382
N, MWA	ppr	ja	0.533	0.486	0.508
		nein	0.355	0.324	0.339
	ppr_w2w	ja	0.488	0.445	0.466
		nein	0.404	0.368	0.385
Adj., Adv., V, N, MWA (N)	ppr	ja	0.549	0.500	0.523
		nein	0.364	0.331	0.347
	ppr_w2w	ja	0.501	0.457	0.478
		nein	0.414	0.377	0.395
N, MWA (N)	ppr	ja	0.545	0.496	0.519
		nein	0.359	0.327	0.342
	ppr_w2w	ja	0.499	0.454	0.475
		nein	0.417	0.379	0.397
WordNet 3.1 MFS	-	-	0.463	0.422	0.442

wurden. Obwohl die Bedeutungsauflösung durch die Fehler der MWA-Erkennung schlechtere Ergebnisse erzielt, ist ein F₁-Wert von 0.523 mit der Wissensbasis WordNet und ein F₁-Wert von 0.470 mit der Wissensbasis DBpedia dennoch akzeptabel. Dies wird vor allem deutlich, wenn man sieht, das andere „state of the art“ Systeme wie Babelfy ähnliche, oder sogar schlechtere Ergebnisse erzielen.

8 Zusammenfassung und Ausblick

Diese Arbeit befasst sich im Rahmen des INDIRECT-Projektes mit der Bedeutungsauflösung unter Berücksichtigung von Multiwort-Ausdrücken (MWAs) bei Anforderungsspezifikationen. Hierfür wurde ein System entwickelt, dass bei einem Eingabetext zunächst mithilfe eines einfach verketteten Zufallsfeldes die MWAs erkennt und anschließend eine wissensbasierte Bedeutungsauflösung mit den Wissensquellen DBpedia und WordNet 3.1 durchführt. Hierfür wurde zunächst untersucht, was der Begriff „Multiwort-Ausdruck“ bedeutet, welche Formen MWAs annehmen können und was der Begriff „Multiwort-Ausdruck“ im Kontext dieser Arbeit umfasst. Anschließend wurden verschiedene Ansätze zur Erkennung von MWAs analysiert und schließlich ein Ansatz des maschinellen Lernens gewählt. Aufgrund dessen wurde aus den Anforderungsspezifikationen mehrerer Datensätze aus der Anforderungsdomäne ein Trainingsdatensatz erstellt. Auf diesem wurde ein einfach verkettetes Zufallsfeld trainiert. Zusätzlich wurde die MWA-Erkennung durch die Suche von Adjektiv+Nomen Kombinationen in einem Wörterbuch, welches für die Wissensbasen erstellt wurde, erweitert. Abschließend wurde ein Verfahren für die Bedeutungsauflösung unter Berücksichtigung von MWAs entwickelt. Da keine Anforderungsdatensätze für das Training von überwachten Ansätzen vorhanden sind und die von unüberwachten Ansätzen erzeugten Ergebnisse aufgrund der geringen Menge an Informationen im Rahmen des INDIRECT-Projektes nur schlecht verwendet werden können, wurde ein wissensbasierter Ansatz basierend auf dem state of the art System „UKB“ gewählt. Für UKB wurde eine neue Datengrundlage geschaffen, indem verschiedene DBpedia Datensätze, auf ihre Anwendbarkeit im Bereich der graphbasierten Bedeutungsauflösung analysiert wurden. Da UKB ein graphbasiertes System ist, wurden diese Analyseergebnisse verwendet, um mehrere Graphen aus den DBpedia Datensätzen zu erzeugen, welche sich für die Bedeutungsauflösung eignen. Zusätzlich wurde ein neuer WordNet 3.1 Graph erzeugt. Des Weiteren wurde ein Verfahren entwickelt, welches in Kombination von UKB verwendet werden kann, um für MWAs, deren vollständige Bedeutung nicht in der Wissensbasis vorhanden ist, eine Teilbedeutung anhand der einzelnen Bestandteile zu bestimmen.

Für die Evaluation des Systems wurde der bereits erstellte Trainingsdatensatz als Musterlösung mit Bedeutungen für die Wissensbasen WordNet 3.1 und DBpedia annotiert. Zunächst wurde eine intrinsische Evaluation der MWA-Erkennung durchgeführt. Da der Trainingsdatensatz auch gleichzeitig als Evaluationsdatensatz agieren soll, wurde eine zehnfache Kreuzvalidierung bei der MWA-Erkennung verwendet. Dies ergab einen F1-Wert von bis zu 0.81. Anschließend wurde eine intrinsische Evaluation der Bedeutungsauflösung durchgeführt, indem alle MWAs und Wörter des Evaluationsdatensatzes extrahiert und

anschließend durch das UKB-System disambiguiert wurden. Die Evaluation wurde dabei unterteilt in eine strenge Evaluation, bei der Teilbedeutungen für MWAs nicht akzeptiert werden und eine Evaluation, bei der Teilbedeutungen akzeptiert werden. Die besten Ergebnisse unter den DBpedia Graphen erzeugte der Hypernyme + Kategorien Graph, der einen F_1 -Wert von 0.387 (ohne Teilbedeutungen) und 0.496 (mit Teilbedeutungen) erreichte. Die beste Konfiguration des WordNet 3.1 Graphen erzielte jedoch bessere Ergebnisse mit einem F_1 -Wert von 0.442 (ohne Teilbedeutungen) und 0.547 (mit Teilbedeutungen). Um das Zusammenspiel zwischen MWA-Erkennung und Bedeutungsauflösung zu testen, wurde auch eine kombinierte Evaluation durchgeführt, bei der sowohl die MWA-Erkennung, als auch die Bedeutungsauflösung zusammenarbeiten. Die besten Ergebnisse erreichte erneut die bereits genannte Konfiguration des WordNet 3.1 Graphen mit einem F_1 -Wert von 0.523. Auch der bereits genannte DBpedia Graph erreichte erneut die besten Ergebnisse mit einem F_1 -Wert von 0.470.

Das Hauptaugenmerk dieser Arbeit lag vor allem darauf eine Bedeutungsauflösung zu implementieren, die die Wissensquelle DBpedia verwendet und dabei MWAs berücksichtigt. Die MWA-Erkennung erzielt dabei bereits gute Ergebnisse, was jedoch auch an der Homogenität des Trainings- und Evaluationsdatensatzes liegt. Mögliche Verbesserungen sind die Verwendung eines Trainingsdatensatzes mit einem breiteren Spektrum an verschiedenen MWAs, um so auch Randfälle besser abzudecken. Bei der Erstellung eines DBpedia Graphen wurde der Fokus vor allem darauf gelegt, die Struktur des WordNet Graphen zu imitieren, da dieser zuvor bereits gute Ergebnisse lieferte. Der DBpedia Datensatz ist jedoch groß und bietet viele verschiedene weitere Möglichkeiten Graphen zu erstellen, die möglicherweise bessere Ergebnisse erzielen und nicht notwendigerweise die Struktur des WordNet Graphen widerspiegeln. Des Weiteren wurden die Gewichte für die einzelnen DBpedia Einträge, anhand der absoluten Häufigkeiten im Wikipedia Korpus erstellt. Ein weiterer interessanter Ansatz, wäre also zu prüfen, ob andere Gewichtungen möglicherweise bessere Ergebnisse für die Bedeutungsauflösung erzielen.

Literaturverzeichnis

- [AAC05] ARRANZ, Victoria ; ATSERIAS, Jordi ; CASTILLO, Mauro: Multiwords and word sense disambiguation. In: *International Conference on Intelligent Text Processing and Computational Linguistics* Springer, 2005, S. 250–262 (zitiert auf den Seiten 17, 25, 27 und 28).
- [ABK⁺07] AUER, Sören ; BIZER, Christian ; KOBILAROV, Georgi ; LEHMANN, Jens ; CYGANIAK, Richard ; IVES, Zachary: DBpedia: A Nucleus for a Web of Open Data. In: ABERER, Karl (Hrsg.) ; CHOI, Key-Sun (Hrsg.) ; NOY, Natasha (Hrsg.) ; ALLEMANG, Dean (Hrsg.) ; LEE, Kyung-Il (Hrsg.) ; NIXON, Lyndon (Hrsg.) ; GOLBECK, Jennifer (Hrsg.) ; MIKA, Peter (Hrsg.) ; MAYNARD, Diana (Hrsg.) ; MIZOGUCHI, Riichiro (Hrsg.) ; SCHREIBER, Guus (Hrsg.) ; CUDRÉ-MAUROUX, Philippe (Hrsg.): *The Semantic Web*. Berlin, Heidelberg : Springer, 2007 (Lecture Notes in Computer Science). – ISBN 978–3–540–76298–0, S. 722–735 (zitiert auf Seite 34).
- [ACC⁺02] ANTONIOL, G. ; CANFORA, G. ; CASAZZA, G. ; DE LUCIA, A. ; MERLO, E.: Recovering traceability links between code and documentation. In: *IEEE Transactions on Software Engineering* 28 (2002), Oktober, Nr. 10, S. 970–983. <http://dx.doi.org/10.1109/TSE.2002.1041053>. – DOI 10.1109/TSE.2002.1041053. – ISSN 1939–3520. – Conference Name: IEEE Transactions on Software Engineering (zitiert auf Seite 1).
- [ALS18] AGIRRE, Eneko ; LACALLE, Oier Lopez d. ; SOROA, Aitor: The risk of sub-optimal use of Open Source NLP Software: UKB is inadvertently state-of-the-art in knowledge-based WSD. In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Melbourne, Australia : Association for Computational Linguistics, Juli 2018, 29–33 (zitiert auf den Seiten 17, 28, 29, 37 und 64).
- [AS09] AGIRRE, Eneko ; SOROA, Aitor: Personalizing PageRank for Word Sense Disambiguation. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece : Association for Computational Linguistics, März 2009, 33–41 (zitiert auf den Seiten 16, 33, 36, 44, 60 und 62).
- [aso20] ASOROA: *Ukb: graph-based WSD and similarity*. <https://github.com/asorua/ukb>. Version: Januar 2020. – original-date: 2012-06-27T07:56:22Z (zitiert auf den Seiten 44 und 46).
- [BS16] BJÖRNE, Jari ; SALAKOSKI, Tapio: UTU at SemEval-2016 Task 10: Binary Classification for Expression Detection (BCED). In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California : Association for Computational Linguistics, Juni 2016, 925–930 (zitiert auf den Seiten 15, 25, 26 und 27).

- [CHCC03] CLELAND-HUANG, J. ; CHANG, C.K. ; CHRISTENSEN, M.: Event-based traceability for managing evolutionary change. In: *IEEE Transactions on Software Engineering* 29 (2003), September, Nr. 9, S. 796–810. <http://dx.doi.org/10.1109/TSE.2003.1232285>. – DOI 10.1109/TSE.2003.1232285. – ISSN 2326–3881 (zitiert auf Seite 42).
- [CHMLP07] CLELAND-HUANG, Jane ; MAZROUEE, Sepideh ; LIGUO, Huang ; PORT, Dan: *nfr*. <http://dx.doi.org/10.5281/zenodo.268542>. Version: März 2007. – type: dataset (zitiert auf Seite 42).
- [coe] *CoEST Datensätze*. <http://www.coest.org/>. – zugegriffen: 2020-01-30 (zitiert auf Seite 42).
- [CRV16] CORDEIRO, Silvio ; RAMISCH, Carlos ; VILLAVICENCIO, Aline: UFRGS&LIF at SemEval-2016 Task 10: Rule-Based MWE Identification and Predominant Supersense Tagging. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California : Association for Computational Linguistics, Juni 2016, 910–917 (zitiert auf den Seiten 14, 15, 26 und 27).
- [CS18] CHAPLOT, Devendra S. ; SALAKHUTDINOV, Ruslan: Knowledge-based Word Sense Disambiguation using Topic Models. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018 (zitiert auf den Seiten 15, 28 und 29).
- [DBp] *DBpedia pre-release*. <https://databus.dbpedia.org/dbpedia/collections/pre-release-2019-08-30>. – zugegriffen: 2020-02-10 (zitiert auf den Seiten 33, 34 und 45).
- [DCLT19] DEVLIN, Jacob ; CHANG, Ming-Wei ; LEE, Kenton ; TOUTANOVA, Kristina: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *arXiv:1810.04805 [cs]* (2019), Mai. <http://arxiv.org/abs/1810.04805>. – arXiv: 1810.04805 (zitiert auf Seite 18).
- [dcm] *DCMI: DCMI Metadata Terms*. <https://dublincore.org/specifications/dublin-core/dcmi-terms/>. – zugegriffen: 2020-03-28 (zitiert auf Seite 34).
- [dim] *POS tags*. <https://universaldependencies.org/docs/en/pos/all.html>. – zugegriffen: 2020-04-03 (zitiert auf den Seiten xiii und 42).
- [HDS06] HAYES, Jane H. ; DEKHTYAR, Alex ; SUNDARAM, Senthil K.: Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods. In: *IEEE Trans. Software Eng.* 32 (2006), Nr. 1, 4–19. <http://dx.doi.org/10.1109/TSE.2006.3>. – DOI 10.1109/TSE.2006.3 (zitiert auf Seite 42).
- [Hey19] HEY, Tobias: INDIRECT: Intent-Driven Requirements-to-Code Traceability. In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, S. 190–191. – ISSN: 2574-1926 (zitiert auf den Seiten 2 und 11).
- [HHD09] HOLBROOK, E. A. ; HAYES, Jane H. ; DEKHTYAR, Alex: Toward Automating Requirements Satisfaction Assessment. In: *2009 17th IEEE International Requirements Engineering Conference*, 2009, S. 149–158. – ISSN: 2332-6441 (zitiert auf Seite 42).
- [HSL16] HOSSEINI, Mohammad J. ; SMITH, Noah A. ; LEE, Su-In: UW-CSE at SemEval-2016 Task 10: Detecting Multiword Expressions and Supersenses using Double-Chained Conditional Random Fields. In: *Proceedings of the*

- 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California : Association for Computational Linguistics, 2016, 931–936 (zitiert auf den Seiten 13, 14, 27, 37 und 41).
- [HSQH19] HUANG, Luyao ; SUN, Chi ; QIU, Xipeng ; HUANG, Xuanjing: Gloss-BERT: BERT for Word Sense Disambiguation with Gloss Knowledge. In: *arXiv:1908.07245 [cs]* (2019), November. <http://arxiv.org/abs/1908.07245>. – arXiv: 1908.07245 (zitiert auf den Seiten 18 und 30).
- [IV98] IDE, Nancy ; VÉRONIS, Jean: Introduction to the special issue on word sense disambiguation: the state of the art. In: *Computational linguistics* 24 (1998), Nr. 1, S. 2–40 (zitiert auf Seite 8).
- [JM19] JURAFSKY, Dan ; MARTIN, James H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2019 <https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf> (zitiert auf den Seiten xiii, 3, 4, 5, 6, 7, 8 und 9).
- [KKKS17] KHURANA, Diksha ; KOLI, Aditya ; KHATTER, Kiran ; SINGH, Sukhdev: Natural language processing: State of the art, current trends and challenges. In: *arXiv preprint arXiv:1708.05148* (2017) (zitiert auf Seite 3).
- [Kli15] KLIEGR, Tomáš: Linked hypernyms: Enriching DBpedia with Targeted Hypernym Discovery. In: *Journal of Web Semantics* 31 (2015), März, 59–69. <http://dx.doi.org/10.1016/j.websem.2014.11.001>. – DOI 10.1016/j.websem.2014.11.001. – ISSN 1570–8268 (zitiert auf Seite 33).
- [Les86] LESK, Michael: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: *Proceedings of the 5th annual international conference on Systems documentation* Citeseer, 1986, S. 24–26 (zitiert auf Seite 9).
- [LIJ⁺15] LEHMANN, Jens ; ISELE, Robert ; JAKOB, Max ; JENTZSCH, Anja ; KONTOKOSTAS, Dimitris ; MENDES, Pablo N. ; HELLMANN, Sebastian ; MORSEY, Mohamed ; KLEEF, Patrick van ; AUER, Sören ; BIZER, Christian: DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. In: *Semantic Web* 6 (2015), Nr. 2, 167–195. <http://dx.doi.org/10.3233/SW-140134>. – DOI 10.3233/SW-140134. – ISSN 15700844 (zitiert auf den Seiten 2, 7, 8 und 33).
- [MAPS18] MELO, Dirce Fernandes d. ; ANTONELLI PONTI, Moacir ; SPRINGER INTERNATIONAL PUBLISHING: *Machine Learning: a Practical Approach on the Statistical Learning Theory*. 2018. – ISBN 978–3–319–94988–8. – OCLC: 1053821192 (zitiert auf Seite 8).
- [McC09] MCCARTHY, Diana: Word sense disambiguation: An overview. In: *Language and Linguistics compass* 3 (2009), Nr. 2, S. 537–558 (zitiert auf den Seiten 4, 7, 8, 9 und 10).
- [Mil95] MILLER, George A.: WordNet: a lexical database for English. In: *Communications of the ACM* 38 (1995), Nr. 11, S. 39–41 (zitiert auf den Seiten 2, 6, 7, 32 und 33).
- [MMWB05] MILES, Alistair ; MATTHEWS, Brian ; WILSON, Michael ; BRICKLEY, Dan: SKOS Core: Simple knowledge organisation for the Web. In: *International Conference on Dublin Core and Metadata Applications* 0 (2005), September, Nr. 0, 3–10. <https://dcpapers.dublincore.org/pubs/article/view/798>. – ISSN 1939–1366. – Number: 0 (zitiert auf Seite 34).

- [MRN14] MORO, Andrea ; RAGANATO, Alessandro ; NAVIGLI, Roberto: Entity Linking meets Word Sense Disambiguation: a Unified Approach. In: *Transactions of the Association for Computational Linguistics* 2 (2014), 231–244. <https://www.aclweb.org/anthology/Q14-1019> (zitiert auf Seite 17).
- [Nak13] NAKOV, Preslav: On the interpretation of noun compounds: Syntax, semantics, and entailment. In: *Natural Language Engineering* 19 (2013), Juli. <http://dx.doi.org/10.1017/S1351324913000065>. – DOI 10.1017/S1351324913000065 (zitiert auf Seite 36).
- [noa20] *Data for the DiMSUM shared task at SEMEVAL 2016*. <https://github.com/dimsum16/dimsum-data>. Version: Januar 2020. – original-date: 2015-09-14T14:37:26Z (zitiert auf Seite 41).
- [PABP17] PELEVINA, Maria ; AREFYEV, Nikolay ; BIEMANN, Chris ; PANCHENKO, Alexander: Making Sense of Word Embeddings. In: *arXiv:1708.03390 [cs]* (2017), August. <http://arxiv.org/abs/1708.03390>. – arXiv: 1708.03390 (zitiert auf den Seiten 19 und 32).
- [PL02] PANTEL, Patrick ; LIN, Dekang: Discovering word senses from text. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* ACM, 2002, S. 613–619 (zitiert auf Seite 10).
- [PMR⁺17] PANCHENKO, Alexander ; MARTEN, Fide ; RUPPERT, Eugen ; FARALLI, Stefano ; USTALOV, Dmitry ; PONZETTO, Simone P. ; BIEMANN, Chris: Unsupervised, Knowledge-Free, and Interpretable Word Sense Disambiguation. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (2017), 91–96. <http://dx.doi.org/10.18653/v1/D17-2016>. – DOI 10.18653/v1/D17-2016. – arXiv: 1707.06878 (zitiert auf Seite 19).
- [PRDB17] PAPANDREA, Simone ; RAGANATO, Alessandro ; DELLI BOVI, Claudio: SupWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Copenhagen, Denmark : Association for Computational Linguistics, September 2017, 103–108 (zitiert auf den Seiten 18 und 30).
- [PRF⁺17] PANCHENKO, Alexander ; RUPPERT, Eugen ; FARALLI, Stefano ; PONZETTO, Simone P. ; BIEMANN, Chris: Unsupervised does not mean uninterpretable : the case for word sense induction and disambiguation. In: *15th Conference of the European Chapter of the Association for Computational Linguistics : proceedings of conference, volume 1: Long Papers*. Stroudsburg, PA : Association for Computational Linguistics, 2017. – ISBN 978-1-945626-34-0, 86–98 (zitiert auf Seite 20).
- [Ram12] RAMISCH, Carlos: A generic framework for multiword expressions treatment: from acquisition to applications. In: *Proceedings of ACL 2012 Student Research Workshop*, Association for Computational Linguistics, 2012, S. 61–66 (zitiert auf Seite 41).
- [Ram15] RAMISCH, Carlos: Multiword expressions acquisition. In: *Springer* 1 (2015), S. 2 (zitiert auf Seite 41).
- [RCCN17] RAGANATO, Alessandro ; CAMACHO-COLLADOS, Jose ; NAVIGLI, Roberto: Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In: *Proceedings of the 15th Conference of the European Chapter*

- of the Association for Computational Linguistics: Volume 1, Long Papers.* Valencia, Spain : Association for Computational Linguistics, 2017, 99–110 (zitiert auf den Seiten 16 und 17).
- [RPS15] RANJAN PAL, Alok ; SAHA, Diganta: Word Sense Disambiguation: A Survey. In: *International Journal of Control Theory and Computer Modeling* 5 (2015), Juli, Nr. 3, 1–16. <http://dx.doi.org/10.5121/ijctcm.2015.5301>. – DOI 10.5121/ijctcm.2015.5301. – ISSN 23194138, 22491155 (zitiert auf den Seiten 9 und 10).
- [SBB⁺02] SAG, Ivan A. ; BALDWIN, Timothy ; BOND, Francis ; COPESTAKE, Ann ; FLICKINGER, Dan: Multiword Expressions: A Pain in the Neck for NLP. In: GELBUKH, Alexander (Hrsg.): *Computational Linguistics and Intelligent Text Processing*. Berlin, Heidelberg : Springer, 2002 (Lecture Notes in Computer Science). – ISBN 978-3-540-45715-2, S. 1–15 (zitiert auf den Seiten xiii, 22 und 23).
- [SHJC16] SCHNEIDER, Nathan ; HOVY, Dirk ; JOHANNSEN, Anders ; CARPUAT, Marine: SemEval-2016 Task 10: Detecting Minimal Semantic Units and their Meanings (DiMSUM). In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California : Association for Computational Linguistics, Juni 2016, 546–559 (zitiert auf den Seiten 13, 14, 15, 25, 26, 27, 41 und 57).
- [wora] *WordNet | A Lexical Database for English*. <https://wordnet.princeton.edu/>. – zugegriffen: 2020-04-04 (zitiert auf Seite 34).
- [Worb] *WordNet Search - 3.1*. <http://wordnetweb.princeton.edu/perl/webwn>, . – zugegriffen: 2020-12-15 (zitiert auf Seite 9).
- [WTLB18] WHITE, Lyndon ; TOGNERI, Roberto ; LIU, Wei ; BENNAMOUN, Mohammed: *Neural Representations of Natural Language*. Bd. 783. Springer, 2018 (zitiert auf den Seiten 4 und 6).
- [Zen19] ZENDEL, Oliver: *WordNet v3.0 vs. v3.1 mapping*. <https://github.com/ozendelait/wordnet-to-json>, 2019 (zitiert auf Seite 65).

Anhang

A Evaluationsergebnisse

Tabelle A.1: Vergleich der kombinierten Ergebnisse (DBpedia/ohne Teilbedeutungen/mit Fehlern der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision	Ausbeute	F1-Wert
Hypernyme + Kategorien	ppr	ja	0.212	0.197	0.204
		nein	0.277	0.257	0.267
	ppr_w2w	ja	0.382	0.355	0.368
		nein	0.354	0.329	0.341
Hypernyme	ppr	ja	0.340	0.309	0.324
		nein	0.365	0.331	0.347
	ppr_w2w	ja	0.297	0.270	0.283
		nein	0.259	0.235	0.246
Wikipedia MFS	-	-	0.274	0.252	0.263

Tabelle A.2: Vergleich der kombinierten Ergebnisse (DBpedia/mit Teilbedeutungen/ohne Fehler der Wissensbasis)

Graph	PageRank	Häufigkeit	Präzision	Ausbeute	F1-Wert
Hypernyme + Kategorien	ppr	ja	0.278	0.329	0.301
		nein	0.359	0.426	0.390
	ppr_w2w	ja	0.488	0.578	0.529
		nein	0.450	0.533	0.488
Hypernyme	ppr	ja	0.425	0.492	0.456
		nein	0.456	0.527	0.489
	ppr_w2w	ja	0.382	0.442	0.410
		nein	0.334	0.387	0.359
Wikipedia MFS	-	-	0.324	0.380	0.350

Tabelle A.3: Vergleich der kombinierten Ergebnisse (WordNet/ohne Teilbedeutungen/mit Fehlern der Wissensbasis)

Kontextwortarten	PageRank	Häufigkeit	Präzision	Ausbeute	F1-Wert
Adj., Adv., V, N, MWA	ppr	ja	0.435	0.397	0.415
		nein	0.293	0.268	0.280
	ppr_w2w	ja	0.400	0.365	0.382
		nein	0.335	0.306	0.320
N, MWA	ppr	ja	0.432	0.394	0.412
		nein	0.290	0.265	0.277
	ppr_w2w	ja	0.404	0.369	0.386
		nein	0.340	0.310	0.324
Adj., Adv., V, N, MWA (N)	ppr	ja	0.445	0.405	0.424
		nein	0.295	0.269	0.281
	ppr_w2w	ja	0.409	0.372	0.390
		nein	0.343	0.312	0.327
N, MWA (N)	ppr	ja	0.441	0.401	0.420
		nein	0.291	0.265	0.277
	ppr_w2w	ja	0.409	0.372	0.390
		nein	0.346	0.315	0.330
WordNet 3.1 MFS	-	-	0.385	0.352	0.368

Tabelle A.4: Vergleich der kombinierten Ergebnisse (WordNet/mit Teilbedeutungen/ohne Fehler der Wissensbasis)

Kontextwortarten	PageRank	Häufigkeit	Präzision	Ausbeute	F1-Wert
Adj., Adv., V, N, MWA	ppr	ja	0.536	0.561	0.548
		nein	0.358	0.375	0.366
	ppr_w2w	ja	0.488	0.511	0.499
		nein	0.400	0.419	0.409
N, MWA	ppr	ja	0.533	0.558	0.545
		nein	0.355	0.372	0.363
	ppr_w2w	ja	0.488	0.511	0.499
		nein	0.404	0.423	0.413
Adj., Adv., V, N, MWA (N)	ppr	ja	0.549	0.574	0.561
		nein	0.364	0.380	0.372
	ppr_w2w	ja	0.501	0.524	0.512
		nein	0.414	0.433	0.423
N, MWA (N)	ppr	ja	0.545	0.569	0.557
		nein	0.359	0.375	0.367
	ppr_w2w	ja	0.499	0.521	0.510
		nein	0.417	0.435	0.426
WordNet 3.1 MFS	-	-	0.463	0.485	0.474