

Semantic Annotation and Search: Bridging the Gap between Text, Knowledge and Language

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
(Dr.-Ing.)

bei der Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte
DISSERTATION

von

Dipl.-Inform. Lei Zhang

Tag der mündlichen Prüfung: 25. April 2017

Referent: Prof. Dr. Rudi Studer

Korreferent: Prof. Dr. Wolfgang Nejdl

To my family.

Abstract

In recent years, the ever-increasing quantities of entities in large knowledge bases on the Web, such as DBpedia, Freebase and YAGO, pose new challenges but at the same time open up new opportunities for intelligent information access. These knowledge bases (KBs) have become valuable resources in many research areas, such as natural language processing (NLP) and information retrieval (IR). Recently, almost every major commercial Web search engine has incorporated entities into their search process, including Google's Knowledge Graph, Yahoo!'s Web of Objects and Microsoft's Satori Graph/Bing Snapshots. The goal is to bridge the *semantic gap* between natural language text and formalized knowledge.

Within the context of globalization, multilingual and cross-lingual access to information has emerged as an issue of major interest. Nowadays, more and more people from different countries are connecting to the Internet, in particular the Web, and many users can understand more than one language. While the diversity of languages on the Web has been growing, for most people there is still very little content in their native language. As a consequence of the ability to understand more than one language, users are also interested in Web content in other languages than their mother tongue. There is an impending need for technologies that can help in overcoming the *language barrier* for multilingual and cross-lingual information access.

In this thesis, we face the overall research question of how to allow for *semantic-aware* and *cross-lingual* processing of Web documents and user queries by leveraging knowledge bases. With the goal of addressing this complex problem, we provide the following solutions: (1) *semantic annotation* for addressing the semantic gap between Web documents and knowledge; (2) *semantic search* for coping with the semantic gap between keyword queries and knowledge; (3) the exploitation of *cross-lingual semantics* for overcoming the language barrier between natural language expressions (i.e., keyword queries and Web documents) and knowledge for enabling cross-lingual semantic annotation and search. We evaluated these solutions and the results showed advances beyond the state-of-the-art. In addition, we implemented a framework of cross-lingual semantic annotation and search, which has been widely used for cross-lingual processing of media content in the context of our research projects.

Acknowledgements

This thesis is the result of my work as a research associate at the Institute AIFB at Karlsruhe Institute of Technology (KIT), Germany. The successful completion of this thesis would not have been possible without the support and guidance of many people.

First, I would like to express my gratitude to my advisor Prof. Dr. Rudi Studer for giving me the opportunity to do this research, and for the freedom, the trust and the support he granted me throughout my Ph.D. study. In addition, I am grateful to the dissertation committee, Prof. Dr. Wolfgang Nejdl and Prof. Dr. York Sure-Vetter, for their valuable feedback.

I would also like to thank the phenomenal team at the Institute AIFB, where they provided an incredibly friendly and supportive atmosphere to make me feel so lucky to be part of this team. In particular, I am thankful to Dr. Achim Rettinger, who supported me during my work on this thesis and in the research projects. Special thanks also go to Dr. Duc Thanh Tran, who introduced me to research work in the first place during his time at the Institute AIFB. Further, I am grateful to many colleagues with whom I enjoyed very much working for many years. Also, I would like to thank all my students who helped me a lot with their ideas and development work as thesis students or assistants at the Institute AIFB.

Finally, I am deeply indebted to my friends and beloved family. Especially, I thank my parents Qingchun Zhang and Chunxiang Xu, who have always supported and believed in me. Most of all, I thank my wife Jingjing He for her continued love, understanding and encouragement. Last but not least, I thank my daughter Siyu to whom I dedicate this work.

Acknowledgements

Contents

Abstract	v
Acknowledgements	vii
I. Foundations	1
1. Introduction	3
1.1. Challenges and Tasks	5
1.2. Research Questions	6
1.3. Contributions of the Thesis	9
1.4. Guide to the Reader	11
2. Basics	13
2.1. Knowledge Bases	13
2.2. Semantic Annotation	15
2.3. Semantic Search	17
2.3.1. Semantic Data Retrieval	18
2.3.2. Semantic-based Information Retrieval	19
II. Semantic Annotation	21
3. Collective Context-Aware Entity Disambiguation	23
3.1. Introduction	23
3.2. Overview	24
3.3. Contextual Entity Detection	27
3.4. Disambiguation Graph Construction	28
3.4.1. Node Weighting	29
3.4.2. Edge Weighting	30
3.5. Collective Entity Disambiguation	30
3.5.1. Eigenvector Centrality	31
3.5.2. PageRank	32
3.5.3. HITS	32

3.6. Experiments	33
3.6.1. Experimental Setup	36
3.6.2. Evaluation Results	37
3.7. Related Work	39
3.8. Conclusions	40
4. Salient Entity Linking	43
4.1. Introduction	43
4.2. Overview	44
4.3. Features and Measures	46
4.3.1. General Features	46
4.3.2. Salient Features	48
4.4. Graph Model and Algorithm	48
4.5. Experiments	50
4.5.1. Experimental Setup	51
4.5.2. Evaluation Results	51
4.6. Related Work	52
4.7. Conclusions	53
III. Semantic Search	55
5. Time-Aware Entity Recommendation	57
5.1. Introduction	57
5.2. Overview	60
5.2.1. Probabilistic Model	60
5.2.2. Data Sources	61
5.2.3. Candidate Selection	62
5.3. Model Parameter Estimation	63
5.3.1. Popularity Model	63
5.3.2. Temporality Model	63
5.3.3. Relatedness Model	64
5.3.4. Mention Model	66
5.3.5. Context Model	66
5.4. Experiments	67
5.4.1. Experimental Setup	68
5.4.2. Results of Entity Retrieval	69
5.4.3. Results of Entity Ranking	70
5.5. Related Work	72
5.6. Conclusions	73

6. Query Rewriting for Keyword Search on Graphs	75
6.1. Introduction	75
6.2. Overview	77
6.2.1. Keyword Search on Graph Data	77
6.2.2. Keyword Query Rewriting	78
6.3. Probabilistic Query Rewriting	80
6.3.1. Probabilistic Model	80
6.3.2. Probabilistic Token Rewriting	81
6.3.3. Probabilities of Query Rewrites	81
6.3.4. Probabilities of Valid Query Rewrites	83
6.3.5. Reward Maximization Framework	86
6.4. Computing Top-k Query Rewrites	86
6.4.1. Indexing	87
6.4.2. Holistic Top-k Query Rewriting	88
6.4.3. Context-based Top-k Query Rewriting	89
6.4.4. Algorithm	91
6.5. Experiments	92
6.5.1. Experimental Setup	92
6.5.2. Efficiency of Query Rewriting	94
6.5.3. Effectiveness of Query Rewriting	95
6.5.4. Impact on Efficiency of Keyword Search	98
6.5.5. Impact on Effectiveness of Keyword Search	100
6.5.6. Analysis of Impact of Query Rewriting	101
6.6. Related Work	102
6.7. Conclusions	103
IV. Cross-lingual Semantics	105
7. Cross-lingual Linked Data Lexica	107
7.1. Introduction	107
7.2. Methodology	108
7.2.1. The Structures in Wikipedia	108
7.2.2. Extraction Process	110
7.3. Datasets	115
7.4. Related Work	116
7.4.1. Dictionary Datasets	117
7.4.2. Lexical Knowledge Bases	117
7.5. Conclusions	118

8. Cross-lingual Keyword Query Interpretation	119
8.1. Introduction	119
8.2. Overview	121
8.3. Query Graph Scoring	123
8.3.1. Key Term Set Score	123
8.3.2. Entity Matching Score	124
8.3.3. Query Entity Graph Score	125
8.4. Top-k Query Graph Exploration	126
8.5. Experiments	129
8.5.1. Effectiveness Evaluation	130
8.5.2. Efficiency Study	132
8.6. Related Work	133
8.7. Conclusions	134
9. A Framework of Cross-lingual Semantic Annotation and Search	137
9.1. Introduction	137
9.2. Cross-lingual and Cross-modal Processing of Media Streams	138
9.3. Cross-lingual Semantic Annotation	140
9.3.1. System Architecture	140
9.3.2. Functionality Description	141
9.4. Cross-lingual and Cross-modal Semantic Search	144
9.4.1. System Architecture	144
9.4.2. Functionality Description	146
9.5. Related Work	149
9.6. Conclusions	150
V. Conclusions	151
10. Conclusions	153
10.1. Summary	153
10.2. Outlook	155
Bibliography	157

List of Figures

1.1. The Linked Open Data (LOD) Cloud. Each node stands for a single data source and each edge connecting two data sources represents the links between them.	4
2.1. An overview of semantic annotation tasks.	16
2.2. An overview of semantic search tasks.	18
3.1. Total processing time (s) of 8 variants of our approach.	38
4.1. Salient entity linking framework.	45
4.2. Example of graph-based disambiguation utilizing a topic-sensitive model. . .	49
5.1. Examples of the candidate query entities and related entities for the user query “Germany Brazil” and the given time range “July 2014”.	58
6.1. Example data graph fragments from different sources covering three domains, i.e., baseball, cars and computer science.	77
6.2. Segment s_{i+1} induced by action α_i performed on segment s_i (set of segments $\mathcal{P}_i(N)$) and token t_{i+1}	84
6.3. Approaches to query rewriting.	89
6.4. Evaluation results for efficiency of query rewriting.	94
6.5. Evaluation results for effectiveness of query rewriting.	97
6.6. Evaluation results for efficiency of keyword search.	99
6.7. Evaluation results for effectiveness of keyword search.	101
7.1. Examples of interlingual resources in Wikipedia. The connecting arrows represent cross-language links between Wikipedia articles in different languages.	108
8.1. Example QEGs generated by our system for the queries (a) “WM Götze”, (b) “online companies of US NDX”, (c) “Google ” and (d) “eBay ”.	122
8.2. Experimental results of query interpretation effectiveness.	131
8.3. Experimental results of query interpretation efficiency.	133
9.1. The xLiMe architecture.	138

List of Figures

9.2. The xLiMe annotation data model.	139
9.3. The system architecture of <i>X-LiSA</i>	141
9.4. Example of annotation service for Web pages.	142
9.5. Example of annotation service output in XML.	143
9.6. Example of annotated social media text in RDF (Turtle format).	143
9.7. Example of SPARQL query.	144
9.8. Example of SPARQL query results.	144
9.9. The system architecture of <i>XKnowSearch!</i>	145
9.10. Example of retrieved news articles for query “ <i>boris johson</i> ”.	147
9.11. Example of retrieved social media posts for query “ <i>boris johson</i> ”.	148
9.12. Example of retrieved TV segments for query “ <i>boris johson</i> ”.	148

List of Tables

3.1.	Examples of the entity disambiguation task, where <i>input mentions</i> and <i>contextual mentions</i> in the given text are highlighted and shadowed, respectively.	25
3.2.	POS patterns in regular expressions, where symbols *, +, and • denote any number of occurrences, one or more occurrences, alternation and concatenation, respectively; NN: singular noun; NNP: proper singular noun; NNS: plural noun; NNPS: proper plural noun; CD: cardinal digit; JJ: adjective; JJS: superlative adjective; JJR: comparative adjective; VBG: present participle of verb; VBN: past participle of verb; CC: conjunction; IN: preposition; POS: possessive 's or '.	28
3.3.	Features of the datasets, including the numbers of documents and ground truth entities as well as the average numbers of ground truth entities and words per document.	33
3.4.	Comparison of 8 variants of our approach and 14 state-of-the-art approaches on 9 datasets using Micro F1 (best results formatted in bold), where if a system provides no results or errors, we report them as N/A (not available).	34
3.5.	Comparison of 8 variants of our approach and 14 state-of-the-art approaches on 9 datasets using Macro F1 (best results formatted in bold), where if a system provides no results or errors, we report them as N/A (not available).	35
4.1.	Statistics of Reuters-128 entity salience dataset.	51
4.2.	The experimental results of salient entity linking.	52
5.1.	Examples of information needs.	68
5.2.	$nDCG@k$ of retrieved entities (with the best results in bold).	70
5.3.	$Recall@k$ of temporally related entities (with the best results in bold).	70
5.4.	The gold-standard ranking of 10 entities (with dynamically related ones in bold) for the query “Germany Brazil” and the date range “July 2014” as well as the rankings by the baseline <i>BSL2</i> and our method with <i>Full Model</i>	71
5.5.	Spearman rank correlation between the gold-standard ranking and the ranking generated by different methods (with the best results in bold).	71
5.6.	Spearman rank correlation between the gold-standard ranking and the ranking by our <i>Full Model</i> for different λ (with the best results in bold).	72
6.1.	Possible query rewrites.	78

List of Tables

6.2.	The extended n -gram index capturing segments containing no more than $N+1$ tokens, and their connections.	87
6.3.	Dataset size, number of relations and tuples, index size/indexing time w.r.t. token index I_{Token} (same one used by all approaches) and the additional indexes used by two variants of our approach I_{PQR} , I_{PVQR} and the one used by the state-of-the-art baseline I_{BQR} (all sizes and time are in MB and minutes). . .	93
6.4.	Number of queries $ Q $, range in number of query keywords $ q $ and relevant results $ R $, average number of query keywords $ \bar{q} $ and relevant results $ \bar{R} $ per query.	93
6.5.	MRR of PVQR vs. η ($\beta = 0.33$).	96
6.6.	MRR of PVQR vs. β ($\eta = 1$).	96
6.7.	The respective effects of our probabilistic model and additional heuristics on effectiveness of query rewriting.	98
7.1.	Statistics about words and labels in Wikipedia.	113
7.2.	Statistics of our datasets and DBpedia NLP Datasets.	115
7.3.	Examples of top-5 results from our datasets and DBpedia NLP datasets for English DBpedia entity <code>dbpedia:Michael_Jordan</code>	115

Part I.
Foundations

1. Introduction

With over one trillion pages and billions of users, the Web is one of the most successful artifacts ever created. From 1 website in 1991 to over 1 billion in 2014¹, the Web has become a *global document repository*, which encompasses practically every topic of human interest. As the founding language, English has always dominated the Web, where it is estimated that 55.5% of all Web content is in English². However, the share of English Web pages decreases and that of other languages increases rapidly, which ensures the multilingual viability of the Web. Accessing documents on the multilingual Web has become an everyday behavior of almost every Web user. The usefulness of Web document access can be seen from the fact that in December 2012, it was noted that 11,8 billion searches were conducted each month on Google³. In consequence, the areas of *Natural Language Processing (NLP)* (Jurafsky & Martin, 2009) and *Information Retrieval (IR)* (Manning et al., 2008) evolved in parallel, which are concerned with capturing the information contained in natural language documents to support their automatic processing and to satisfy information needs of users.

With the goal of extending the existing Web by bringing semantics to its content, the Semantic Web community has come a long way since its beginnings in the late 1990s and early 2000s. There has been an increasing effort in which the community has envisioned how semantics and the Web can be combined. By adding a multitude of language standards and software components, the Semantic Web can better enable humans and machines to work in cooperation (Berners-Lee et al., 2001). Over the last 10 years, there has been a growing amount of research on interaction paradigms that allow users to profit from the expressive power of Semantic Web standards while at the same time hiding their complexity behind an intuitive and easy-to-use interface. Linked Open Data (LOD)⁴ is such a way of publishing semantic data on the Web that gives humans and machines direct access to such semantic data (Bizer et al., 2009a; Heath & Bizer, 2011). In addition, there has been recent work on practical design consideration of publishing multilingual linked data (Buitelaar & Cimiano, 2014), which is now continued by the W3C community group on Best Practices for Multilingual Linked Open Data⁵. It is important to note that many LOD sources are generally in multiple languages. As an exam-

¹<http://www.internetlivestats.com/total-number-of-websites/>

²<https://blog.unbabel.com/2015/06/10/top-languages-of-the-internet/>

³<https://www.comscore.com/Insights/Press-Releases/2013/1/comScore-Releases-December-2012-US-Search-Engine-Rankings>

⁴<http://lod-cloud.net/>

⁵<https://www.w3.org/community/bpmlod/>

1. Introduction

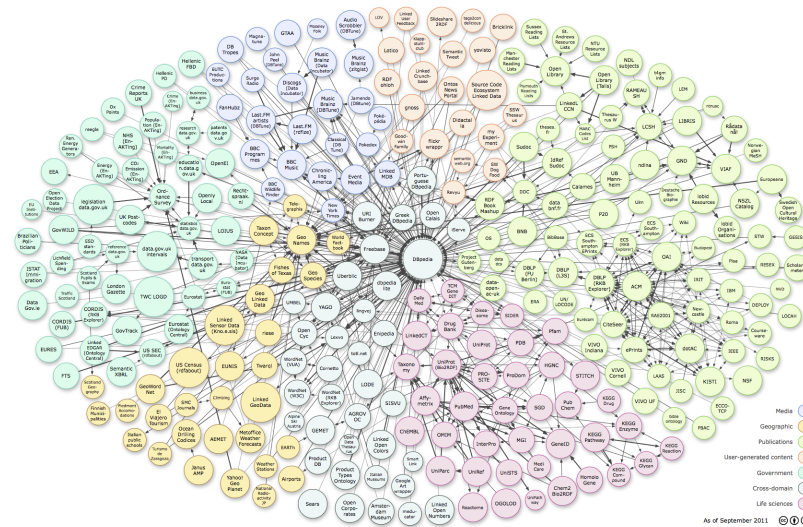


Figure 1.1.: The Linked Open Data (LOD) Cloud. Each node stands for a single data source and each edge connecting two data sources represents the links between them.

ple shown in Fig. 1.1, DBpedia⁶, staying in the center of the LOD cloud, is a crowd-sourced community effort to extract structured knowledge from multilingual Wikipedia, resulting in localized versions of DBpedia in more than 100 languages, and to make this information available on the Web (Auer et al., 2007; Bizer et al., 2009b).

Since its invention in 1989 by Tim Berners-Lee, the Web, which was originally designed as a *global document repository*, has radically altered the way that information is shared by lowering the barrier to publishing and accessing documents. On the other hand, the ever-increasing quantities of semantic data in large knowledge bases (KBs), such as DBpedia, Freebase and YAGO, pose new challenges but at the same time open up new opportunities of intelligent information access on the Web. These knowledge bases contain a vast amount of entities and the knowledge about the entities such that the Web also serves as a *global knowledge repository* of entities. In recent years, many research activities involving entities have emerged, such as entity disambiguation and linking in natural language text as well as entity retrieval and recommendation for a given information need. In addition, almost every major commercial Web search engine has announced their work on incorporating entity knowledge from structured knowledge bases into the search process, including Google's Knowledge Graph, Yahoo!'s Web of Objects and Microsoft's Satori Graph / Bing Snapshots.

In this thesis, we are concerned with how to connect different kinds of Web items including documents, queries and knowledge, also in the multilingual and cross-lingual settings.

⁶<http://dbpedia.org/>

1.1. Challenges and Tasks

Given the Web repository of both documents and knowledge, two major challenges are faced when accessing information on the Web. We introduce the first challenge as follows:

Challenge 1: Semantic Gap. Accessing both documents and knowledge on the Web can be efficient when user information needs are expressed as keyword queries. However, Web documents and keyword queries are usually treated as plain text by current search engines. In other words, term-based matching algorithms are used to retrieve the results according to a given information need. This results in problems for ambiguous terms. For example, “*Paris*” can denote the capital of France, towns in Canada and USA, or the socialite and heiress *Paris Hilton*. Moreover, it is not feasible to satisfy complex information needs with the term-based retrieval paradigm. For example, given the information need of finding publications of all researchers from AIFB expressed by the keyword query “*publications AIFB*”, current Web search engines cannot directly provide the answers, for which users have to first search and browse to find all researchers at AIFB and then another round of search and browsing is needed to find information about their publications. Therefore, there exists a *semantic gap* between the ambiguous and vague formulation in natural language and its semantic representation in the form of entities and their relations from knowledge bases.

In order to bridge the *semantic gap* between natural language expressions and their formal knowledge representations, we introduce two tasks that will be addressed in this thesis:

Task 1.1: Semantic Annotation. The process of tying natural language text and semantic models together is generally referred to as *semantic annotation* (Bontcheva & Cunningham, 2011), which can be characterized as the dynamic construction of interrelationships between unstructured documents and structured knowledge. It helps to bridge the ambiguity of natural language text when expressing their computational representation in the formal knowledge. More specifically, semantic annotation is about attaching additional semantic information to unstructured documents through metadata that is referring to resources in structured knowledge bases, such as entities as the main focus in this thesis.

Task 1.2: Semantic Search. The topic formed around the use of semantics for various search tasks is usually known as *semantic search*, which tries to offer users more precise and relevant results by using semantics that is frequently encoded in knowledge bases. Semantic search has been studied by researchers in several communities from different viewpoints (Tran et al., 2011; Bontcheva et al., 2013; Bast et al., 2016). In this thesis, we focus on using knowledge about entities and their relationships explicitly given in structured knowledge bases to provide relevant answers for information needs of users expressed by keyword queries. Another task of semantic search we are concerned with in the context of Information Retrieval (IR), also known as *Semantic-based IR*, is to retrieve Web documents on the basis of relevance to entities from knowledge bases instead of the term-based retrieval paradigm.

Besides the semantic gap, another challenge we face for cross-lingual access to information

on the Web is stated below:

Challenge 2: Language Barrier. Within the context of globalization, cross-lingual access to information has emerged as an issue of major interest. Nowadays, more and more people from different countries are connecting to the Internet and many Web users are able to understand more than one language. For example, more than half of the citizens in the European Union can speak at least one other language than their mother tongue⁷. While the diversity of languages on the Web has been growing in recent years, for most people there is still very little content in their native language. As a consequence, multilingual users probably formulate the information needs using their native language, but they are interested in relevant information in any language they can understand. With the goal that users from all countries have access to the same information on the Web, there exists a *language barrier* for cross-lingual access to information originally produced for a different culture and language.

In order to address both the challenges of *semantic gap* and *language barrier*, we introduce the cross-lingual extensions of the above two tasks in the following:

Task 2.1: Cross-lingual Semantic Annotation. Semantic annotation are typically language dependent, which aims to link unstructured documents in one language with structured knowledge grounded in the same language. *Cross-lingual semantic annotation* goes beyond the general task, as it faces annotation across the boundary of languages, where the documents to be annotated and the resources in knowledge bases used for annotation are in different languages.

Task 2.2: Cross-lingual Semantic Search. *Cross-lingual semantic search* extends the task of semantic search in the monolingual setting in the sense that users can use keyword queries in any language for finding relevant answers in knowledge bases grounded in any other languages and for retrieving multilingual documents, especially in the languages different from the query language. In addition, multilingual users could issue queries consisting of keywords in multiple languages and specifying query languages should not be the burden of users, which makes cross-lingual semantic search more challenging.

Concerned with these two major challenges and the corresponding tasks, in the next section we will formulate the overall research question, which leads to several individual research questions according to different challenges and tasks.

1.2. Research Questions

The principal research question of this thesis is:

How to allow for semantic-aware and cross-lingual processing of Web documents and user queries by leveraging knowledge bases?

⁷http://ec.europa.eu/public_opinion/archives/ebs/ebs_237.en.pdf

This broad research question is broken down into eight specific research questions, each of which entails a combination of different challenges and tasks as stated above and will be addressed in the remainder of this thesis.

The first two research questions are derived from Challenge 1 *Semantic Gap* and concerns Task 1.1 *Semantic Annotation*:

Research Question 1. *How to enable context-aware and collective entity disambiguation for different types of input mentions in documents?*

The increasing amount of entities in large knowledge bases can help to bridge unstructured text with structured knowledge and the key is to disambiguate entity mentions in text with entities in knowledge bases. Recently, many methods have been proposed to tackle this problem. However, most of them assume certain characteristics of the given input mentions, e.g., only named entities or individual words are considered. In this regard, the research question of how to enable context-aware and collective entity disambiguation for different types of input mentions will be investigated in Chapter 3.

Research Question 2. *How to enable salient entity discovery in documents?*

For many entity-centric applications, entity salience for a document has become a very important factor. This raises an impending need to identify a set of salient entities that are central to the given input document. With respect to this issue, we introduce a new task of salient entity linking with the focus on the disambiguation of entity mentions into salient entities in a document that existing solutions to entity linking cannot well address. This research question will be investigated in Chapter 4.

The next two research questions are derived from Challenge 1 *Semantic Gap* and concerns Task 1.2 *Semantic Search*:

Research Question 3. *How to enable time-aware entity recommendation for temporal information needs?*

There has been an increasing effort to develop techniques for related entity recommendation, where the task is to retrieve a ranked list of related entities given a keyword query. Another trend in information retrieval (IR) is to take temporal aspects of a given query into account when assessing the relevance of documents. However, while this has become an established functionality in document search engines, the significance of time has not yet been recognized for entity recommendation. In this regard, we address this gap by introducing the task of time-aware entity recommendation. This research question will be investigated in Chapter 5.

Research Question 4. *How to enable effective and efficient keyword search on knowledge graphs?*

Keyword search on graph data has attracted large interest. Using keyword queries, users can search for complex structured results from knowledge graphs. Existing work so far focuses on the efficient processing of keyword queries or effective ranking of results. In addition, recent

work studies the problem of keyword query cleaning. The motivation is keyword queries are dirty, often containing words that are misspelled or words that do not directly appear but are semantically equivalent to words in the data. Besides dirty queries, keyword search solutions also face the problem of search space explosion, i.e., the space of possible results is generally exponential in the number of query keywords. These issues will be studied in Chapter 6.

Besides Challenge 1 *Semantic Gap*, the remaining four research questions are also derived from Challenge 2 *Language Barrier* and concern Task 2.1 *Cross-lingual Semantic Annotation* and Task 2.2 *Cross-lingual Semantic Search*, respectively:

Research Question 5. *How to allow for an easy mapping of natural language expressions in different languages to entities in knowledge bases?*

Recently, multilingual and cross-lingual access to information on the Web has drawn increasing attention. It is essential to propose new technologies that can help with scaling the traditionally monolingual tasks to multilingual and cross-lingual applications. In order to enable cross-lingual semantic annotation and search, there is a clear need for cross-lingual groundings of entities to allow for an easy mapping of natural language expressions in different languages to entities in knowledge bases. This research question will be investigated in Chapter 7.

Research Question 6. *How to enable cross-lingual keyword query interpretation?*

As a simple and intuitive paradigm of expressing information needs of users, keyword queries have enjoyed widespread usage, but suffer from the challenges including ambiguity, incompleteness and cross-linguality. More specifically, keyword queries are naturally ambiguous and incomplete, i.e., keywords could refer to different things in different contexts and only aliases, acronyms and misspellings are usually given in the queries. In addition, keyword queries might be formulated in one language or even multiple languages by multilingual users, and they are interested in relevant information in any language that they can understand. These challenges will be addressed in Chapter 8.

Research Question 7. *How to enable cross-lingual entity linking in multilingual documents?*

The previous research questions concerning semantic annotation are limited to the monolingual setting. However, for certain entities, their information is only available in knowledge bases grounded in a foreign language. To address this issue, we consider a new task of cross-lingual entity linking, where input documents are in a different language than that used for describing entities in knowledge bases. This technology is crucial for many entity-centric applications in a cross-lingual context. Ultimately, the goal is to construct cross-lingual entity linking tools that can link words or phrases in unstructured text in one language to entities in structured knowledge bases grounded in any other languages. This research question will be investigated in Chapter 9.

Research Question 8. *How to enable entity-based cross-lingual information retrieval (IR) by exploiting knowledge bases?*

Due to an increasing portion of queries involving entities for Web document search (Pound et al., 2010), the exploitation of *entities and their relations* in knowledge bases beyond the term-based paradigm for information retrieval (IR) has become an area of particular interest. In addition, the recent progress in cross-lingual technologies is largely due to the increased availability of multilingual data sources. Based on that, the research question of how to enable entity-based cross-lingual IR will be investigated in Chapter 9.

With regard to the above research questions, this thesis provides several novel contributions that we will outline in the next section.

1.3. Contributions of the Thesis

This thesis comprises eight main contributions, each of which results from the investigation of one specific research question. In the following we briefly describe each contribution, which will be detailed in its own chapter.

Contribution 1. *Context-aware and collective disambiguation of entities in documents*

Based on our publication (Zhang et al., 2016b), we present a context-aware approach to collective entity disambiguation of the input mentions with different characteristics in a consistent manner in Chapter 3. The main contribution includes the contextual entity detection based on a set of predefined part-of-speech (POS) tag patterns, which provides the context to help with entity disambiguation for the given input mentions, and the collective disambiguation using a class of algorithms for estimating the relative importance of candidate entities in the constructed disambiguation graph based on Markov chains. Through the extensive experiments, we show that our approach outperforms the state-of-the-art methods in most cases.

Contribution 2. *A topic-sensitive model for salient entity linking in documents*

In order to tackle the new problem of salient entity linking, we propose a graph-based entity linking framework, which integrates several features including prior mention importance, mention-entity compatibility, entity-entity coherence and in particular a topic-sensitive model capturing entity-category association and document-specific category importance. We have experimentally shown that our approach achieves a significant improvement over the baselines. The evaluation results also show that the topic-sensitive model indeed helps with the salient entity discovery. We have discussed this contribution in our previously published paper (Zhang et al., 2015b) and present a revised version in Chapter 4.

Contribution 3. *A probabilistic model for time-aware entity recommendation*

Based on our publication (Zhang et al., 2016d), we propose a statistically sound probabilistic model to tackle the novel task of time-aware entity recommendation in Chapter 5. We decompose the task into several well defined probability distributions reflecting heterogeneous entity knowledge and show how all parameters of our probabilistic model can be effectively

estimated solely based on data sources publicly available on the Web. Due to the lack of existing benchmark datasets for this challenge, we have created new datasets to enable empirical evaluation and the evaluation results show that our proposed approach considerably improves the performance compared to time-agnostic approaches.

Contribution 4. *A probabilistic method of query rewriting for effective and efficient keyword search on knowledge graph*

Towards a query rewriting solution that enables more effective and efficient keyword search on graph data, we propose a novel approach to probabilistic ranking and context-based computation of query rewrites. In addition, we investigate the impacts of our ranking mechanism and computation algorithm for query rewriting on both effectiveness and efficiency of keyword search, respectively. Based on our publication (Zhang et al., 2013), we show that our approach to query rewriting is several times faster than the state-of-the-art baseline and also yields higher quality of rewrites especially for large datasets. Most importantly, we show that these improvements on query rewriting also carry over to the actual keyword search. This contribution will be presented in Chapter 6.

Contribution 5. *Cross-lingual linked data lexica*

Based on our publications (Zhang et al., 2014a; Zhang et al., 2014b), we present our cross-lingual linked data lexica in Chapter 7. With the goal of allowing for an easy mapping of natural language expressions in different languages to entities in knowledge bases, we exploited various kinds of structures in Wikipedia, such as anchor text of hyperlinks and cross-language links, to derive different associations between natural language expressions extracted from Wikipedia editions in multiple languages and linked data resources. We believe that the extracted lexica can help to support many cross-lingual applications using semantic technologies, such as cross-lingual semantic annotation and search.

Contribution 6. *A knowledge base approach to cross-lingual keyword query interpretation*

In order to address the challenges that traditional keyword search systems mainly suffer from, we introduce a knowledge base approach to cross-lingual query interpretation by transforming keywords in different languages to their semantic representation. Based on our publication (Zhang et al., 2016c), we propose a scoring mechanism for effective query interpretation ranking and a top-k graph exploration algorithm for efficient query interpretation generation. Through the empirical evaluation, we show that our ranking mechanism and the top-k graph exploration algorithm lead to a considerable improvement over the baseline methods on both effectiveness and efficiency, respectively. This contribution will be presented in Chapter 8.

Contribution 7. *A system of cross-lingual entity linking*

Most entity linking systems in the monolingual setting rely on context similarity measures based on Bag-of-Words (BOW) models. However, these approaches suffer from the vocabulary mismatch problem for the cross-lingual entity linking task. To address this issue, we use

our cross-lingual lexica, described in Chapter 7, for mention-entity matching and applied a concept-based approach for cross-lingual context similarity calculation (Zhang et al., 2015c) to capture the local mention-entity compatibility. In addition, our approach to graph-based collectively entity disambiguation used by monolingual entity linking (Zhang et al., 2016b) has been adapted to the cross-lingual setting. Based on our publications (Zhang & Rettinger, 2014; Zhang et al., 2017), this contribution will be presented in Chapter 9.

Contribution 8. *A system of entity-based cross-lingual information retrieval (IR)*

Based on our publications (Zhang et al., 2016a; Zhang et al., 2017), we present a novel system of entity-based cross-lingual information retrieval (IR) in Chapter 9. By leveraging entities in multilingual knowledge bases, keyword queries and Web documents in different languages can be captured on their semantic level to avoid the ambiguity of terms and to bridge the language barrier between them. To the best of our knowledge, this is the first entity-based system for multilingual and cross-lingual IR, where users can issue keyword queries in any language, which can even contain keywords in multiple languages, for retrieving documents in any other languages.

The above contributions collectively address the principal research question stated in Section 1.2 and show how to leverage large knowledge bases available on the Web for *semantic-aware* and *cross-lingual* processing of Web documents and user queries.

1.4. Guide to the Reader

This thesis comprises ten chapters, which are divided into five parts according to the addressed tasks. Firstly, Chapter 1 and Chapter 2 provide the foundations for this thesis. Then the following core chapters (Chapter 3 - Chapter 9) cover all the research questions stated before and present our solutions and contributions to each specific research question. Finally, Chapter 10 provides the conclusions of this thesis.

Part I provides the **Foundations** for this thesis.

- **Chapter 1.** We introduce the challenges and tasks concerned in this thesis, break down the principal research question into eight individual research questions, summarize the main contributions, and provide this guide to the reader.
- **Chapter 2.** We provide a brief introduction to knowledge bases and preliminaries for the tasks of semantic annotation and semantic search.

Part II discusses the task of **Semantic Annotation**.

- **Chapter 3.** We show that our approach to entity disambiguation achieves promising results by leveraging the contextual entities derived from the given document and collective algorithms based on Markov chains.
- **Chapter 4.** We propose a new task of salient entity linking and present an approach to this problem by utilizing a topic-sensitive model based on Wikipedia categories.

Part III discusses the task of **Semantic Search**.

- **Chapter 5.** We present the first probabilistic model that takes both relevance and timeliness into consideration for entity recommendation given temporal information needs.
- **Chapter 6.** We show that query rewriting can help to improve not only the result quality but also the runtime performance of keyword search on knowledge graphs.

Part IV discusses the **Cross-lingual** extensions of **Semantic Annotation and Search**.

- **Chapter 7.** We present our cross-lingual linked data lexica constructed by exploiting the multilingual Wikipedia and the linked data resources.
- **Chapter 8.** We present a knowledge base approach to cross-lingual query interpretation by transforming query keywords in different languages to their semantic representation.
- **Chapter 9.** We present a framework of cross-lingual semantic annotation and search by exploiting entities in multilingual knowledge bases, which serve as an interlingua to connect keyword queries and Web documents across languages.

Part V concludes this thesis.

- **Chapter 10.** The thesis ends with a summary of the main conclusions and an outlook on future research directions.

2. Basics

This chapter first briefly introduces some notable knowledge bases and then gives an overview of the foundations, both in the fields of semantic annotation and semantic search.

2.1. Knowledge Bases

Knowledge bases on the Web are a backbone of many intelligent information systems. Comprehensive knowledge bases in machine-readable representations have been a goal of Artificial Intelligence (AI) for decades. Seminal projects, such as Cyc (Lenat, 1995) that manually compiles common sense knowledge and Wordnet (Fellbaum, 1998) that aims to build a lexical knowledge base, yield high-quality repositories of general concepts and relations. These early forms of knowledge bases contain logical statements, such as computer scientists are humans, and all humans have a biological mother and a biological father. However, early knowledge bases like Cyc and WordNet lack knowledge about individual entities of this world and their relations. For example, they do not contain entities like Tim Berners-Lee nor the knowledge that Tim Berners-Lee is a computer scientist and also the inventor of the World Wide Web.

More recently, numerous endeavors have been engaged to overcome the prior limitations of sparse entity coverage and build large-scale knowledge bases, which usually contain millions of individual entities and relations between them. On the other hand, knowledge bases on the Semantic Web are typically provided using Linked Data (Bizer et al., 2009a). Currently, it is the best practice for publishing knowledge in a graph-based representation, e.g., using the Resource Description Framework (RDF), where entities as nodes are connected by relations as edges in the graph (e.g., Tim Berners-Lee is the founder of the World Wide Web Foundation), and entities can have types, denoted by *is a* relations (e.g., Tim Berners-Lee is a computer scientist, the World Wide Web Foundation is an organization). Nowadays, there are 2,740 knowledge bases in the Linked Open Data cloud (Ermilov et al., 2016), such as DBpedia, YAGO and Freebase as the most prominent ones.

There are different ways of constructing such knowledge bases (Paulheim, 2017). For example, they can be manually crafted by an organization or a small group of individuals like Cyc (Lenat, 1995) and WordNet (Fellbaum, 1998), crowd-sourced by a community like Freebase (Bollacker et al., 2008) and Wikidata (Vrandečić & Krötzsch, 2014), automatically extracted from large-scale and semi-structured Web knowledge bases such as Wikipedia, like

DBpedia (Auer et al., 2007) and YAGO (Suchanek et al., 2007), or extracted from unstructured text at Web scale, leading to knowledge bases like NELL (Carlson et al., 2010).

In the following, we give an overview of existing knowledge bases, which have been constructed by different methods.

Cyc and OpenCyc. As one of the oldest knowledge bases of common sense in traditional AI research, Cyc (Lenat, 1995) is a curated knowledge base developed and maintained by the CyCorp company starting in 1984, whose domain is all of human consensus reality, such as the common sense fact that “every tree is a plant”. Since Cyc is proprietary, a smaller version of the knowledge base called OpenCyc is publicly available both as downloadable OWL ontologies as well as via Semantic Web endpoints.

WordNet WordNet (Fellbaum, 1998) is a lexical knowledge base for the English language developed at the Cognitive Science Laboratory of Princeton University starting in 1985. It groups different types of words, i.e., nouns, verbs, adjectives and adverbs, into sets of cognitive synonyms, called *synsets*, each expressing a distinct concept. Words with multiple meanings, namely ambiguous words, can belong to multiple synsets. Synsets are interlinked by means of semantic relations, such as hypernymy (subclass-of) and meronymy (part-of).

Freebase. Freebase (Bollacker et al., 2008) is a public knowledge base created through crowdsourcing. Since curating a universal knowledge base containing all possible entities of this world is infeasible for most individuals and organizations, Freebase took a different way from the curated knowledge bases like Cyc and WordNet. It provided an interface that allowed end-users to contribute to the knowledge base by editing structured data with schema templates for most kinds of possible entities, such as persons, organizations, movies and so on.

Wikidata. Like Freebase, Wikidata (Vrandečić & Krötzsch, 2014) is also a collaboratively edited knowledge base with community effort operated by the Wikimedia foundation starting in 2012. As its particularity, Wikidata contains not only facts but also the provenance metadata (e.g., the source and date) about such facts, so that their validity can be checked. After the shutdown of Freebase, its data is subsequently integrated into Wikidata.

DBpedia. DBpedia (Auer et al., 2007) is the most popular and prominent knowledge base in the LOD cloud. It is extracted from structured information contained in Wikipedia, such as from infobox boxes, categorization information, geo-coordinates and external links. DBpedia has been extensively used in various research areas, especially in the Semantic Web community. Due to its role as the hub of LOD, DBpedia also contains many links to other knowledge bases in the LOD cloud such as Freebase, OpenCyc, GeoNames, DBLP and so on.

YAGO. Like DBpedia, YAGO (Suchanek et al., 2007) is also due to the great success of Wikipedia and algorithmic advances in information extraction. It comprises knowledge extracted from Wikipedia (e.g., infoboxes and categories), WordNet (e.g., synsets and hyponymy) and GeoNames. While DBpedia creates different interlinked knowledge bases for each Wikipedia language edition, YAGO aims at an automatic fusion of knowledge extracted

from various language editions of Wikipedia using different heuristics. In addition, the use of WordNet as a taxonomic backbone also makes YAGO different from DBpedia.

NELL. While DBpedia and YAGO mainly rely on semi-structured information, methods for extracting knowledge bases from unstructured data have been also proposed. NELL (Carlson et al., 2010) is one of the earliest systems that attempt to perform two tasks everyday: (1) extracting facts from text found in a large-scale corpus of Web pages (e.g., `PLAYINSTRUMENT(GEORGE_HARRISON, GUITAR)`); (2) learning to improve the competence of the system to extract more facts from the Web, more accurately. NELL has been running 24 hours/day since January 2010 and is still running today, continuously extending its knowledge base.

2.2. Semantic Annotation

Annotation, or tagging, is in general about attaching additional information to a document or a selected part of the document. It provides metadata about an existing piece of unstructured text. Natural language processing (NLP) is one of the most commonly used techniques for annotation of text by adding linguistic tags. Compared with such linguistic tagging, semantic annotation goes one level deeper by enriching the unstructured text with a context that is further linked to resources in knowledge bases. It helps to bridge the ambiguity of natural language text when expressing their computational representation in the formal knowledge.

Semantic annotation can be performed manually, automatically or semi-automatically. While manual semantic annotation can only be feasible in very limited domains and applications or through crowd-sourcing platforms on the Web, it is in general too expensive to carry out without any automation. Semi-automatic semantic annotation has been mainly used for post-editing and correcting the results of the automatic methods by human annotators. A number of manual and semi-automatic annotation systems are described in (Bontcheva & Cunningham, 2011). In this thesis, we focus on automatic semantic annotation.

An overview of tasks involved in semantic annotation are shown in Fig. 2.1. As for most semantic annotation tasks, a linguistic analysis is usually needed for text preprocessing. In this case, the typical analysis includes part-of-speech (POS) tagging that adds POS tags to text, named entity recognition and classification (NERC) that identifies mentions of named entities (NE) in text and labels them with their types, and dependency parsing that even derives entire parse trees from text. Such linguistic information can help with deriving semantics from text. According to different kinds of semantic resources, the main tasks carried out during semantic annotation include word sense disambiguation, entity linking and disambiguation as well as relation extraction, where the task of *entity linking and disambiguation* is the focus of thesis. In the following, we briefly introduce all these three tasks:

Word Sense Disambiguation. Word sense disambiguation (WSD) is a historical task in the fields of natural language processing (NLP) and artificial intelligence (AI). Over the past few

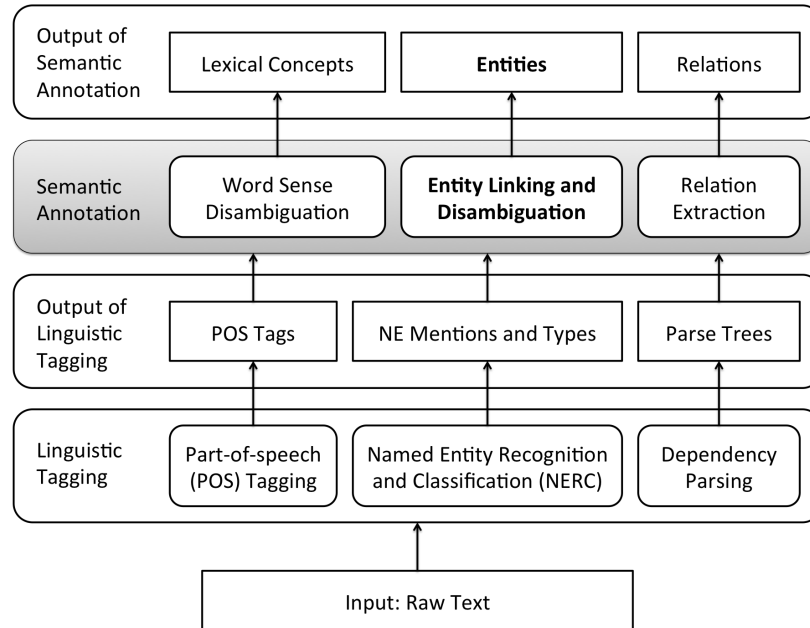


Figure 2.1.: An overview of semantic annotation tasks.

decades, a large body of work had been done in word sense disambiguation. The task aims to determine the meaning of each word in an input text. This requires large-scale lexical resources containing senses for all words in a given language. One such lexical knowledge base is WordNet (Fellbaum, 1998) as described in Sec. 2.1, which has been widely used for research in word sense disambiguation because it represents nearly all senses of words in the English language as clearly defined synsets. A comprehensive survey covering details of word sense disambiguation can be found in (Navigli, 2009).

Entity Linking and Disambiguation. As large knowledge bases of individual entities became available, it enabled the linking of words or phrases in natural language text to entities in knowledge bases. The challenges of entity linking lie in entity recognition and entity disambiguation. The first stage, i.e., entity recognition, is to determine which word sequences in text might refer to an entity, for which usually no knowledge base is required. The task is only to identify possible entity mentions, which typically relies on the linguistic processing of text, such as POS tagging and named entity recognition. The second stage, i.e., entity disambiguation aims at mapping ambiguous entity mentions onto canonical entities like persons, organizations or movies in knowledge bases such as DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007) and Freebase (Bollacker et al., 2008). This task is similar to word sense disambiguation but with different semantic resources used for annotation, i.e., lexical concepts for word sense disambiguation and entities for entity disambiguation. More details about entity linking and disambiguation can be found in (Shen et al., 2015).

Relation Extraction. Different from entity linking, relation extraction aims at extracting subject-predicate-object triples from natural language text, where the subject and object are grounded to an entity in a given knowledge base, and the predicate is grounded to a relation in an ontology / schema of the knowledge base. This problem has been well studied in the research area of information extraction (IE) (Sarawagi, 2008), where a fixed set of predefined relations to be extracted are usually assumed. Considering an example for extracting triples for the place of birth relation from a given sentence “Berners-Lee was born in London”, such a triple would be <Tim_Berners-Lee, place_of_birth, London>, where both subject and object as well as the predicate denote the corresponding entities and relation in a knowledge base. Besides this extraction paradigm, there are also proposals for open information extraction (Banko et al., 2007), where the goal is to extract as many triples as possible for any relation from the given text. For the above example sentence, a typical open IE system would extract <Berners-Lee, was born in, London>, where the subject and object, especially the predicate are not grounded in any knowledge base, but are simply expressed using words from the sentence.

2.3. Semantic Search

Search can be considered as an automated process that helps users to effectively find the right information given their information needs, which are typically expressed in the form of keyword queries. In traditional keyword search, the information matching query keywords is retrieved as answers to users. As the opposite, semantic search is about finding information that is not based on the presence of words, but rather on semantics, i.e., the meaning of words.

Semantic search has been studied by researchers in several communities from different viewpoints (Tran et al., 2011; Bontcheva et al., 2013; Bast et al., 2016). The research on semantic search can be classified according to two dimensions: (1) the underlying data including documents (as unstructured text in natural language) and databases / knowledge bases (consisting of structured / semantic data); (2) the query types including keyword queries (containing just a few keywords), structured queries (in a formal language like SQL or SPARQL), and natural language queries (i.e., complete questions as humans typically pose them).

Nowadays, keyword search is still the most ubiquitous search paradigm, which all of the major Web search engines are using. In this thesis, we are concerned with two principal tasks with information needs expressed by keyword queries: (1) one is often referred to as *semantic data retrieval*, where the subjects of interest could be concepts, entities and subgraphs from knowledge bases; (2) the other is *semantic-based information retrieval*, where the goal is to return documents on the Web with the help of semantic data obtained by (1). An overview of semantic search tasks are shown in Fig. 2.1, which will be briefly discussed in the following.

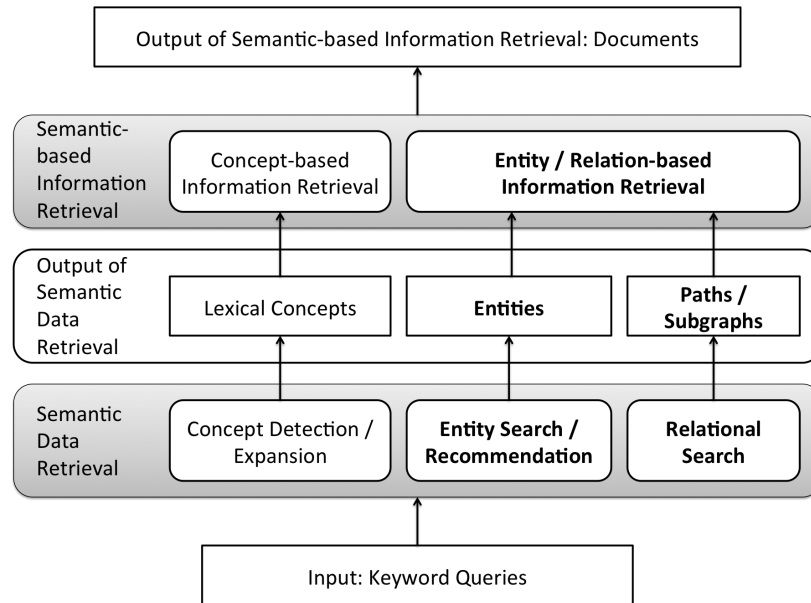


Figure 2.2.: An overview of semantic search tasks.

2.3.1. Semantic Data Retrieval

The main strength of semantic data in knowledge bases is that the information needs can be satisfied with precise semantics. In this regard, semantic data retrieval is to go beyond retrieving a list of documents supported by existing Web search engines to deliver direct answers from knowledge bases. In this section, we briefly introduce three tasks including *concept detection / expansion*, *entity search / recommendation* as well as *relational search*, where the latter two tasks are the focus of this thesis.

Concept Detection / Expansion. With the goal of dealing with semantic ambiguity of words, concept detection in keyword queries aims to retrieve concepts matching keywords from a lexical knowledge base, such as WordNet (Fellbaum, 1998). Compared with the task of word sense disambiguation in long documents, solutions to concept detection in short keyword queries also disambiguate words against concepts but without linguistic information such as POS tags that can be exploited. Concept detection in keyword queries has long history for supporting concept-based information retrieval (Giger, 1988). In addition, the detected concepts can also be expanded by following their relations, which can help with concept-based query expansion (Voorhees, 1994).

Entity Search / Recommendation. Entity search has been defined as finding an entity that is explicitly named in a keyword query, also called query entity, in a knowledge base (Pound et al., 2010). Besides individual entities, semantic search engines such as Falcons (Cheng &

Qu, 2009) and Sig.ma (Tummarello et al., 2010) also return entity descriptions as results by using semantic data available on the Web. A variant of entity search is entity recommendation, where the goal is to rank relationships between a query entity and other entities in a knowledge base (van Zwol et al., 2010; Kang et al., 2011). In the context of Web search, related entity recommendation has been defined as finding the entities related to a query entity appearing in a Web search query (Blanco et al., 2013).

Relational Search. The information needs of relational search goes beyond individual entities. An answer to this type of search usually consists of both entities and their relations, i.e., a path or a subgraph connecting the query entities. Keyword search solutions to relational search have been proposed for dealing with different kinds of data, including XML and relational database (Kacholia et al., 2005; He et al., 2007; Li et al., 2008) as well as RDF graph (Tran et al., 2009). Typically, such solutions first match keywords to entities and then find paths or subgraphs connecting these entities. Relational search enables users to find out how some entities are related to some other entities such that it can help with query interpretation. In order to take advantage of both usability of keyword queries and expressiveness of structured queries, several solutions to query interpretation have been proposed (Zenz et al., 2009; Tran et al., 2009; Demidova et al., 2010; Demidova et al., 2012b), where a keyword query is translated into a ranked list of structured queries such that users can select the ones that represent their information needs.

2.3.2. Semantic-based Information Retrieval

The main challenges in dealing with keyword queries are their *ambiguity* and *variation*, i.e., the same meaning can be expressed in different ways and one keyword can also have different meanings. Semantic-based information retrieval (IR) relies on the semantics of queries to address the challenges of *ambiguity* and *variation*, where the main difference from semantic data retrieval is the focus on using semantics to find documents, rather than forming queries against semantic data in knowledge bases. In this section, we briefly introduce two tasks in this direction, i.e., *concept-based information retrieval* and *entity / relation-based information retrieval*, where the latter one is the focus of this thesis.

Concept-based Information Retrieval. The use of concepts from a lexical knowledge base to deal with *ambiguity* of query keywords has been investigated as the task of concept-based information retrieval (Giger, 1988). In this regard, WordNet (Fellbaum, 1998) has been most commonly used and found to be beneficial in disambiguating query keywords and in choosing their senses (Voorhees, 1993). In addition, concept-based query expansion aims to address the *variation* of keywords by firstly detecting the concept behind the query and then expanding query keywords based on concepts instead of terms (Qiu & Frei, 1993). For example, using WordNet thesaurus to represent concepts, query expansion can be performed by following links between WordNet synonym sets (Voorhees, 1994).

Entity / Relation-based Information Retrieval. The problem of exploiting semantic annotations in information retrieval has been investigated for many years (Alonso & Zaragoza, 2008; Balog et al., 2016). Due to the large-scale knowledge bases available on the Web, most Web search queries are related to entities and relations in such knowledge bases and thus can be mapped to them. With the help of semantic annotation, especially entity linking and relation extraction, as discussed in Sec. 2.2, it enables users to find documents that mention one or more entities or even relations between them by capturing queries and documents at the semantic level. In this way, documents can be retrieved on the basis of relevance to entities and relations from knowledge bases instead of the term-based retrieval paradigm. In addition, it can also bridge the language barriers between queries and documents due to the availability of large multilingual knowledge bases.

Part II.

Semantic Annotation

3. Collective Context-Aware Entity Disambiguation

The rapidly increasing amount of entities in large knowledge bases can help to bridge unstructured text with structured knowledge and thus be beneficial for many entity-centric applications. The key issue is to link entity mentions in text with entities in knowledge bases, where the main challenge lies in mention ambiguity. Many methods have been proposed to tackle this problem. However, most of the methods assume certain characteristics of the input mentions and documents, e.g., only named entities are considered. In this chapter, we propose a context-aware approach to collective entity disambiguation of the input mentions in text with different characteristics in a consistent manner. We extensively evaluate the performance of our approach over 9 datasets and compare it with 14 state-of-the-art methods. Experimental results show that our approach outperforms the existing methods in most cases.

3.1. Introduction

With large repositories of structured knowledge about entities publicly available on the Web, *entity linking* has emerged as a topic of major interest. The challenges of entity linking lie in entity recognition and disambiguation. The first stage, namely *entity recognition*, serves to detect words or phrases in text, also called mentions, that are likely to denote entities; the second stage, namely *entity disambiguation*, performs the disambiguation of mentions into entities, which is the focus of this work. Many methods (Mendes et al., 2011; Han et al., 2011; Ferragina & Scaiella, 2012; van Erp et al., 2013; Usbeck et al., 2014; Rizzo et al., 2014; Piccinno & Ferragina, 2014; Milne & Witten, 2008b) have been proposed to tackle the problem of entity disambiguation, where the goal is to map each input mention given in text to the corresponding entity in knowledge bases. The knowledge base adopted in this work is DBpedia, a crowd-sourced community effort to extract structured information from Wikipedia.

In general, entities can be grouped into named entities and nominal entities. While named entities have proper names, nominal entities do not have a proper name but are referenced typically by a noun phrase, which has a noun as its head word. For instance, given the sentence “US President Barack Obama will land in India for a visit.”, the mentions “*Barack Obama*” and “*India*” refer to the named entities `Barack_Obama` and `India`, while the mentions “*US*

President” and *“visit”* refer to the nominal entities `President_of_the_United_States` and `State_visit`. Recognizing named entities (NER) in natural language text has been extensively addressed (Finkel et al., 2005), where the output is labeled noun phrases representing name entities. However, these are not entities explicitly and uniquely denoted in a knowledge base. Recently, a lot of research has focused on *named entity disambiguation* that goes one step beyond NER, where the task is to disambiguate mentions of named entities in natural language text by linking them to their corresponding entities in a knowledge base (Han et al., 2011; Hoffart et al., 2011; Shen et al., 2012). On the other hand, *word sense disambiguation* (WSD) is a task aimed at assigning meanings to word occurrences within text, where such words usually refer to nominal entities (Navigli, 2009; Navigli, 2012). Some other work focuses on *Wikification*, a task of disambiguating entities in text into their corresponding Wikipedia pages (Mihalcea & Csomai, 2007; Milne & Witten, 2008b; Cheng & Roth, 2013). In addition, a given input mention might not match any entity in the knowledge base. Such mentions are usually defined as *unlinkable* and NIL will be returned. In this work, we do not assume any specific entity types for entity disambiguation, where the entities to be disambiguated could be named entities, nominal entities and unlinkable entities.

The main contributions of this chapter are: (1) the introduction of a context-aware approach to collective entity disambiguation for different kinds of input mentions in text in a consistent manner; (2) the contextual entity detection based on a set of predefined part-of-speech (POS) tag patterns, which provides the context to help with entity disambiguation for the given input mentions; (3) the collective disambiguation using a class of algorithms for estimating the relative importance of candidate entities in the constructed disambiguation graph based on Markov chains; and (4) an extensive evaluation of the performance of our approach over 9 datasets and an empirical comparison with 14 state-of-the-art methods using GERBIL (Usbeck et al., 2015), a general entity annotation benchmark.

The rest of this chapter is organized as follows. We present the overall approach in Sec. 3.2. The details of contextual entity detection and disambiguation graph construction are provided in Sec. 3.3 and Sec. 3.4, respectively. Based on that, we discuss the collective disambiguation using Markov chains in Sec. 3.5. Evaluation results are then presented in Sec. 3.6. Finally, we survey the related work in Sec. 3.7 and conclude in Sec. 3.8.

3.2. Overview

In this section, we first formally formulate the task of entity disambiguation and then briefly describe our approach.

Definition 1 (Entity Disambiguation). *Let $M_I = \{m_1, m_2, \dots, m_p\}$ denote a set of given input mentions in a document D , where each mention m is encoded by an integer pair $\langle p, l \rangle$ with p as the occurrence position of m in D and l as the length of m . Given a knowledge base*

Example 1	<p>Text: The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it.</p> <p>Input mentions: $\{m_1 = \langle 24, 5 \rangle, m_2 = \langle 48, 13 \rangle, m_3 = \langle 75, 4 \rangle, m_4 = \langle 85, 5 \rangle\}$</p> <p>Referent entities of input mentions: $\{m_1.e=\text{Shire_}(Middle\text{-}earth), m_2.e=\text{Frodo_}Baggins, m_3.e=\text{One_}Ring, m_4.e=\text{Bilbo_}Baggins\}$</p>
Example 2	<p>Text: The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it.</p> <p>Input mentions: $\{m_1 = \langle 4, 5 \rangle, m_2 = \langle 41, 6 \rangle, m_3 = \langle 110, 5 \rangle\}$</p> <p>Referent entities of input mentions: $\{m_1.e=\text{Novel}, m_2.e=\text{Hobbit}, m_3.e=\text{Quest}\}$</p>
Example 3	<p>Text: The novel begins in the Shire, where the hobbit Frodo Baggins inherits the Ring from Bilbo and undertakes the quest to destroy it.</p> <p>Input mentions: $\{m_1 = \langle 4, 5 \rangle, m_2 = \langle 48, 13 \rangle, m_3 = \langle 106, 3 \rangle\}$</p> <p>Referent entities of input mentions: $\{m_1.e=\text{Novel}, m_2.e=\text{Bilbo_}Baggins, m_3.e=\text{NIL}\}$</p>

Table 3.1.: Examples of the entity disambiguation task, where *input mentions* and *contextual mentions* in the given text are highlighted and shadowed, respectively.

KB containing a set of entities $E = \{e_1, e_2, \dots, e_n\}$, the task of entity disambiguation is to find a function $\mu : M_I \rightarrow E \cup \{NIL\}$, which maps each input mention m to an entity e in *KB*, denoted by $m.e$, or to *NIL* if the mention cannot be linked to any entity in *KB*.

For each given input mention $m \in M_I$, we first retrieve a set of *candidate entities* E_m using a dictionary collected from different structures in Wikipedia, which contains each pair of entity and *surface form*, i.e., a word or phrase that can be used to refer to the corresponding entity. Then the objective of entity disambiguation is to determine which entity $e \in E_m$ is the most likely entity referred to by m , also called *referent entity*. Besides the given input mentions in M_I for a document D , a set of mentions M_C containing the mentions $m \notin M_I$ in D , called *contextual mentions*, that can refer to some entities in the knowledge base, called *contextual entities*, could also help with the entity disambiguation task. While the input mentions are explicitly given in the task, the contextual mentions have to be derived by our approach, which will be discussed in Sec. 3.3.

Some examples of the entity disambiguation task for different types of input and contextual mentions are shown in Table 3.1. For instance, only the input mentions for named entities are given in Example 1, which corresponds to the typical *named entity disambiguation*. Most existing approaches (Han et al., 2011; Hoffart et al., 2011; Usbeck et al., 2014) to this task take into account only the named entities but ignore the nominal entities, such as **Hobbit** referred to by the contextual mention “hobbit”, which can indeed help with named entity disambiguation since such contextual entities are related to the actual referent entities of the input mentions. In Example 2, some individual words referring to nominal entities are given as input mentions. This is similar to the *word sense disambiguation* task, where the goal is to identify which sense of a word (i.e. meaning) is used in the given text. Based on the lexical knowledge bases, such as WordNet, knowledge-based approaches are able to obtain good performance (Agirre & Soroa, 2009). Instead of lexical knowledge bases, large structured knowledge bases, such

as DBpedia, can also be employed, such that the contextual entities appearing in the given document can be utilized for the disambiguation of word senses as entities in such knowledge bases. In Example 3, three input mentions, i.e., “novel”, “Frodo Baggins” and “the”, are given and the actual referent entities include the nominal entity `Novel`, the named entity `Frodo_Baggins` and `NIL`. As shown in Table 3.1, the contextual entities in the given document can be beneficial to disambiguating all the input mentions. Even for `NIL` corresponding to “the”, which could also refer to some entities according to our dictionary, such as the entity `THE_multiprogramming_system`, the contextual entities can help to return `NIL`, because they are not related to any candidate entities of the input mention “the”.

Besides the above examples, the input mentions for entity disambiguation can be yielded by many other ways, e.g., they can cover only salient entities in the given document annotated based on voter agreement or determined by domain experts. A description of 9 datasets used in our experiments will show different characteristics of the input mentions and documents. In order to address the problem of entity disambiguation for such input mentions and documents in a consistent way, we propose a framework with the following three modules:

- **Contextual Entity Detection.** The entity disambiguation task critically depends on the specific context in a given document D , which is crucial in solving the problem of entity ambiguity. In this module, we propose a new approach to *contextual entity detection* based on a set of predefined POS patterns. The goal is to select contextual entities representing the context of D , which can help to disambiguate the entities for the input mentions. In Example 1 of Table 3.1., if the contextual entity `Hobbit` referred to by “hobbit”, a fictional, diminutive, humanoid race in J. R. R. Tolkien’s fiction, is given, the input mention “Bilbo” should more likely refer to `Bilbo_Baggins`, the character of J. R. R. Tolkien’s fiction, than the band with the same name, namely `Bilbo_Baggins_(band)`.
- **Disambiguation Graph Construction.** By combining the candidate entities of input mentions and the contextual entities detected in the given document, we construct the *disambiguation graph* in this module, which captures both the local mention-entity compatibility and the global entity-entity coherence as its graph structure. In this way, the constructed disambiguation graph allows us to encode different types of dependencies. In Example 1 of Table 3.1, the candidate entity `Bilbo_Baggins` depends on the mention “Bilbo” and is related to the contextual entity `Hobbit`.
- **Collective Entity Disambiguation.** We then consider the *collective entity disambiguation* over the disambiguation graph as a stochastic process based on Markov chains. The intuition is that the actual referent entity of an input mention m should be more relevant in the disambiguation graph in the sense that it tends to have more relations to other candidate entities and contextual entities, than the rest of candidate entities of m , which should have less relations on average and be more isolated. In Example 1 of Table 3.1, the actual referent entity `Bilbo_Baggins` of the input mention “Bilbo” is connected to more candidate entities and contextual entities, such as `One_Ring` and `Hobbit`, than the

other candidate entities of “Bilbo”, such as `Bilbo_Baggins_(band)`.

3.3. Contextual Entity Detection

Given the input document, we need to derive the contextual entities, which can be either named entities or nominal entities. For instance, in Example 3 of Table 3.1, “Bilbo” and “hobbit” can refer to the named entity `Bilbo_Baggins` and the nominal entity `Hobbit` respectively, both of which can help with entity disambiguation for the input mentions. To obtain these contextual entities, it is essential to first detect their mentions.

We firstly present the extraction process of our dictionary used to map surface forms to their corresponding DBpedia entities. We have exploited several structures in Wikipedia. As each Wikipedia article describes an entity in DBpedia, article titles, redirect pages and link anchors in Wikipedia can be used to refer to the corresponding entity. For each DBpedia entity, we extract its surface forms using these sources. Besides that, we also derive the co-occurrence relations between entities and terms, where we utilize the terms that co-occur with an entity in its surrounding sentences in Wikipedia. In addition, the link frequency between each pair of entity and surface form and the co-occurrence frequency between each pair of entity and term are also extracted, which are used for node weighting of the disambiguation graph discussed in Sec. 3.4. More details about the dictionary construction can be found in (Zhang et al., 2014a; Zhang et al., 2014b).

Next we introduce two methods that have been widely used for mention detection based on N-gram and NER, and discuss their limitations, which serve as the motivation of our proposed method based on POS analysis. Some existing work on mention detection (Mihalcea & Csomai, 2007; Milne & Witten, 2008b) firstly gathers all n-grams from the given document and the extracted n-grams matching surface forms of entities are then selected as entity mentions. These methods can detect both named entities and nominal entities but could also generate a lot of noise, i.e., mentions without actual referent entities. Such entities will be considered in the module of collective entity disambiguation, which are not helpful and might even result in degraded performance. In some other work (Hoffart et al., 2011), named entity recognition (NER) has been performed on the input text to detect named entities, which are then used for entity disambiguation and linking. Due to the limitation of selected algorithms and training data, NER systems usually only focus on several types of named entities, e.g., Person, Location and Organization, such that the entities in other types cannot be detected. More importantly, all the nominal entities that might be important contextual entities and be beneficial to entity disambiguation are just ignored.

To address the problems of N-gram and NER methods, we propose a POS tagging based method for detecting mentions of contextual entities. Given the input document D , we firstly perform the POS tagging on D and then extract all sequences conforming to a set of predefined POS patterns, denoted by P , as shown in Table 3.2. These POS patterns are reasonable for

3. Collective Context-Aware Entity Disambiguation

Pattern Name	POS Tag Pattern	Example
Noun 1 (NP1)	(NN NNP NNS NNPS)+	Kobe Bryant, Basketball
Noun 2 (NP2)	NP1 • (CD)+	Windows 10, ISO 8
Noun 3 (NP3)	(CD)+ • NP1	2014 World Cup
Noun (NP)	NP1 NP2 NP3	
Description 1 (DP1)	(JJ JJS JJR)+	Military (Operation)
Description 2 (DP2)	(VBG VBN)+	Judging (Day), Linked (Data)
Description 3 (DP3)	NP3 • POS+	NBA's (Player)
Description (DP)	(DP1 DP2 DP3)	
Compound Noun 1 (CNP1)	DP* • NP	Australian Prime Minister Linked Open Data NBA's All-time Scoring List
Conjunction (CP)	(CC IN)	of, in, and, with
Compound Noun 2 (CNP2)	CNP1 • CP • CNP1	Police in Sweden First Minister of Scotland
Contextual Mention	CNP1 CNP2	

Table 3.2.: POS patterns in regular expressions, where symbols *, +, | and • denote any number of occurrences, one or more occurrences, alternation and concatenation, respectively; NN: singular noun; NNP: proper singular noun; NNS: plural noun; NNPS: proper plural noun; CD: cardinal digit; JJ: adjective; JJS: superlative adjective; JJR: comparative adjective; VBG: present participle of verb; VBN: past participle of verb; CC: conjunction; IN: preposition; POS: possessive 's or '.

mentions of both named entities and nominal entities. Empirical experiments also show that this method could detect entity mentions with a high recall. The extracted sequences based on the POS patterns serve as the mentions of contextual entities, which have to satisfy two conditions: (1) they can refer to some entities in DBpedia based on our dictionary containing the set of surface forms of all entities, denoted by SF ; (2) they are not contained in the set of input mentions M_I . Then we obtain the set of contextual entity mentions M_C as follows

$$M_C = \{m | \forall sq_m \in SQ_D : sq_m \in P \wedge m.s \in SF \wedge m \notin M_I\} \quad (3.1)$$

where SQ_D represents the set of all possible sequences of POS tags generated by performing POS tagging on the given document D , sq_m and $m.s$ denote a sequence of POS tags and an entity surface form w.r.t. the mention m , respectively. Based on our dictionary, we generate the set of contextual entities E_m for each mention $m \in M_C$ and the set of all contextual entities is then just the union of E_m for all mentions in M_C defined as $E_C = \cup_{m \in M_C} E_m$.

3.4. Disambiguation Graph Construction

In this module, we retrieve the set of candidate entities E_m for each input mention $m \in M_I$ based on our dictionary and the set of all candidate entities is defined as $E_I = \cup_{m \in M_I} E_m$. We

then build a directed weighted graph $G = \{V, R\}$, called *disambiguation graph*, where $V = E_I \cup E_C$ is the union of candidate entities and contextual entities, and R is the set of directed edges representing entity relations, where an edge between two entities e_i and e_j will be added into R if the following conditions are satisfied: (1) e_i is linked to e_j in KB, i.e., $e_i \rightarrow e_j$; (2) e_i and e_j have different mentions, i.e., $e_i \in E_m, e_j \in E_{m'}$ and $m \neq m'$. Based on Example 1 of Table 3.1, we show some examples of nodes and edges in the graph. For instance, the input mention “Frodo Baggins” will result in the candidate entity Frodo_Baggins, “Bilbo” will result in some candidate entities, such as Bilbo_Baggins and Bilbo_Baggins_(band), and the contextual mention “hobbit” will result in some contextual entities, such as Hobbit and The_Hobbit_(film_series). All these candidate entities and contextual entities are added into the graph as nodes. Then, the relations between all such entities are added into the graph as edges, such as `characterRace` connecting Frodo_Baggins and Bilbo_Baggins with Hobbit. Our approach then employs several features to assign weights to nodes and edges in G .

3.4.1. Node Weighting

For each mention m , we first calculate its *prior importance* $PI(m)$ that captures how likely the surface form $m.s$ is used as an entity mention as follows

$$PI(m) = \frac{count_{anchor}(m.s)}{count_{anchor}(m.s) + count_{raw}(m.s)} \quad (3.2)$$

where $count_{anchor}(s)$ denotes the number of articles that contain s as anchor text of links and $count_{raw}(s)$ denotes the number of articles where s appears as raw text without links.

For each pair of mention m and its associated entity e , we calculate their semantic similarity $SS(m, e)$ that represents the local *mention-entity compatibility* between m and e as

$$SS(m, e) = \alpha \cdot LP(m, e) + (1 - \alpha) \cdot CS(m, e) \quad (3.3)$$

where $LP(m, e)$ denotes the link probability of e for m and $CS(m, e)$ denotes the context similarity between m and e , α is a tunable parameter. The link probability $LP(m, e)$ captures how likely $m.s$ refers to e , which can be calculated as

$$LP(m, e) = \frac{count_{link}(e, m.s)}{\sum_{e_i \in E_s} count_{link}(e_i, m.s)} \quad (3.4)$$

where $count_{link}(e, s)$ denotes the number of links using s as anchor text pointing to e as destination and E_s is the set of entities that have the surface form s . The context similarity $CS(m, e)$ between m and e can be calculated using cosine similarity on the term vectors of the context of m and e as

$$CS(m, e) = \cos(m.c, e.c) = \frac{\langle m.c, e.c \rangle}{|m.c| \cdot |e.c|} \quad (3.5)$$

3. Collective Context-Aware Entity Disambiguation

where $m.c$ is the frequency vector of terms that are contained in the surrounding sentences of m and $e.c$ is the frequency vector of terms that co-occur with e extracted from Wikipedia.

Using the prior mention importance as the initial evidence and the mention-entity compatibility capturing the most likely entity behind the mention, we calculate the score of each $v \in V$ corresponding to entity e that has the mention m as

$$S(v) = PI(m) \cdot SS(m, e) \quad (3.6)$$

Based on that, the probability $p(v)$ serving as the weight of each node $v \in V$ can be calculated as follows

$$p(v) = \frac{S(v)}{\sum_{u \in V} S(u)} \quad (3.7)$$

3.4.2. Edge Weighting

The module of collective entity disambiguation relies on the global *entity-entity coherence*, which reflects the intuition that entities appearing in the same document are more likely to be related. Therefore, we calculate the semantic relatedness between each pair of connected entities e_i and e_j in G by adopting the Wikipedia link based measure (Milne & Witten, 2008a) as

$$SR(e_i, e_j) = 1 - \frac{\log(\max(|E_i|, |E_j|)) - \log(|E_i \cap E_j|)}{\log(|E|) - \log(\min(|E_i|, |E_j|))} \quad (3.8)$$

where E_i and E_j are the sets of entities that link to e_i and e_j in KB respectively, and E is the set of all entities in KB.

Based on the entity relatedness, we calculate the transition probability for each edge from u to v in G as follows

$$p(v|u) = \begin{cases} \frac{SR(u,v)}{\sum_{w \in OUT_u} SR(u,w)} & \text{if } (u, v) \in R \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

where OUT_u is the set of entity nodes such that for each node $w \in OUT_u$, there is an edge from u to w in G .

3.5. Collective Entity Disambiguation

Based on the constructed disambiguation graph, we consider collective entity disambiguation as a stochastic process, more specifically, a first-order Markov chain model. Intuitively, it can be interpreted as a process where a single ‘‘random walker’’ traverses a graph in a stochastic manner for an infinitely long time and the fraction of time that the walker spends at a single

node, i.e., the stationary distribution of the Markov chain, can then be considered as being proportional to an estimate of the importance of this node relative to others in the graph.

For the disambiguation graph G , where nodes represent both candidate entities and contextual entities in a given document D and edges correspond to relations between these entities, the Markov analogy could be seen as an *ad infinitum* stream of thoughts that refers to the interconnection in a sequence of entities thought by the author for writing the document D .

There is a class of algorithms that have been proposed for estimating relative importance of nodes in a graph based on Markov chains. To address the entity disambiguation problem, we start with the simple method of eigenvector centrality (Bonacich, 1972), and then discuss the well-known PageRank (Brin & Page, 1998) and HITS algorithms (Kleinberg, 1999) as well as their extensions with prior bias (White & Smyth, 2003).

3.5.1. Eigenvector Centrality

Eigenvector centrality (Bonacich, 1972) provides a principled method to combine the importance of a node in a graph with its neighbors in ranking. The scores correspond to the likelihood of arriving in each node by traversing through the graph with a random starting node, where the decision to take a particular path is based on the weighted edges. Given the disambiguation graph G , eigenvector centrality of nodes in G can be defined as the principle eigenvector of the transition matrix \mathbf{T} constructed from the weights of edges in G . The equation of the principle eigenvector \mathbf{c} is defined as $\mathbf{c} = \mathbf{T} \cdot \mathbf{c}$, where the maximal eigenvalue λ corresponding to \mathbf{c} is 1, since \mathbf{T} is a square stochastic adjacency matrix. Each entry $T(u, v)$ in \mathbf{T} specifies the transition probability $p(v|u)$ from node u to v in G , which is defined in Eq. 3.9, and each entry $c(v)$ in \mathbf{c} represents the eigenvector centrality of node v , which is proportional to the sum of eigenvector centrality of all nodes connected to v . It can be estimated through the iterative calculation as

$$c^{i+1}(v) = \sum_{u \in IN_v} p(v|u) \cdot c^i(u) \quad (3.10)$$

where IN_v is the set of entity nodes such that for each node $u \in IN_v$, there is an edge from u to v in G . For each mention m having a set of candidate entities E_m , we choose the entity with the maximal $c(v)$ as the predicted linking entity, i.e., $e_m^* = \arg \max_{v \in E_m} c(v)$.

Based on the Perron-Frobenius theorem (Seneta, 2006), an irreducible and aperiodic Markov chain can be guaranteed to converge to a unique stationary distribution. If a Markov chain has reducible or periodic components, a random walker may get stuck in these components and never visit the other parts of the graph. To solve this problem, PageRank (Brin & Page, 1998) suggests reserving some probability for jumping to any node in the graph, such that the random walker can “escape” from periodic or disconnected components, which makes the graph irreducible and aperiodic. We will discuss this issue in the following section.

3.5.2. PageRank

PageRank (Brin & Page, 1998) is the most well-known example of Markov chains for ranking Web pages in search engine results, where the Markov analogy is defined as a “random surfer” surfing the Web based on the hyperlinks between Web pages. In the traditional PageRank, a uniform probability is assigned to any node in the Web hyperlink graph in case of random jumps of a surfer. Given the disambiguation graph $G = (V, R)$, we first define a $|V| \times 1$ vector \mathbf{p}_V , whose elements are $\frac{1}{|V|}$. With the uniform prior probability $p(v)$ in \mathbf{p}_V attached to each node v and the probability $p(v|u)$ of transitioning from all nodes u linked to v , as defined in Eq. 3.9, the iterative probability equation of v in a Markov chain can be defined as follows

$$\pi^{(i+1)}(v) = (1 - d) \cdot (\sum_{u \in IN_v} p(v|u) \cdot \pi^{(i)}(u)) + d \cdot p(v) \quad (3.11)$$

where IN_v is the set of entity nodes such that for each node $u \in IN_v$, there is an edge from u to v in G and d is the damping factor, which determines how often a surfer jumps back to node v with probability $d \cdot p(v)$ and is typically chosen in the interval $[0.1, 0.2]$.

In (Haveliwala, 2002; Jeh & Widom, 2003), PageRank has been extended to generate “personalized” ranks, called *personalized* PageRank, where the prior probability of nodes are non-uniform such that it can effectively bias the resulting ranks to prefer certain kinds of nodes. In this regard, we replace the uniform distribution $p(v) = \frac{1}{|V|}$ for each $v \in V$ with the non-uniform prior probability $p(v)$ defined in Eq. 3.7. This is analogous to adding a set of weighted outgoing edges for all the nodes in G . Intuitively, this creates a small probability for a random walk to go to some other nodes in G , although it may not have been initially connected to the current node. After convergence of the Markov chain, each node v will achieve a stationary probability $\pi(v)$. For each mention m having a set of candidate entities E_m , we choose the entity with the maximal $\pi(v)$ as the predicted linking entity, i.e., $e_m^* = \arg \max_{v \in E_m} \pi(v)$.

3.5.3. HITS

Besides PageRank, another seminal contribution to ranking nodes in Web hyperlink graph is HITS (Kleinberg, 1999), where two kinds of scores, namely hub and authority, are assigned to nodes in the graph depending on the topology of Web graph. In (White & Smyth, 2003), HITS has been extended by fitting it into a more Markov fashion, where prior probabilities are assigned to nodes to permit random jumps. Given the disambiguation graph G , we incorporate the prior probability vector \mathbf{p}_V for nodes in G into the extended HITS algorithm. Similar to PageRank, the prior probability $p(v)$ in \mathbf{p}_V can be defined as uniform distribution, i.e., $p(v) = \frac{1}{|V|}$, or non-uniform according to Eq. 3.7. This yields the following iterative equation for both hub and authority scores of each node v

$$a^{(i+1)}(v) = (1 - d) \cdot (\sum_{u \in IN_v} \frac{p(v|u) \cdot h^{(i)}(u)}{H^{(i)}}) + d \cdot p(v) \quad (3.12)$$

$$h^{(i+1)}(v) = (1 - d) \cdot (\sum_{u \in OUT_v} \frac{p(u|v) \cdot a^{(i)}(u)}{A^{(i)}}) + d \cdot p(v) \quad (3.13)$$

Datasets	#Doc.	#Ent.	Avg. Ent./Doc.	Avg. Word/Doc.
ACE2004	57	253	4.44	459
AIDA/CoNLL	231	4485	19.42	213
AQUAINT	50	727	14.54	320
DBpedia Spotlight	58	330	5.69	32
IITB	103	11242	109.15	763
KORE50	50	143	2.82	14
MSNBC	20	650	32.5	688
N ³ RSS-500	500	590	1.18	34
N ³ Reuters-128	128	637	4.98	140

Table 3.3.: Features of the datasets, including the numbers of documents and ground truth entities as well as the average numbers of ground truth entities and words per document.

where IN_v (OUT_v) is the set of entity nodes such that for each $u \in IN_v$ ($u \in OUT_v$) there is an edge from u to v (v to u) and d is the damping factor similar to PageRank. $H^{(i)}$ and $A^{(i)}$ are defined as

$$H^{(i)} = \sum_{v \in V} \sum_{u \in IN_v} p(v|u) \cdot h^{(i)}(u) \quad (3.14)$$

$$A^{(i)} = \sum_{v \in V} \sum_{u \in OUT_v} p(u|v) \cdot a^{(i)}(u) \quad (3.15)$$

After convergence of the algorithm, each node v corresponding to a candidate entity gets a hub score $h(v)$ and an authority score $a(v)$. Given the set of candidate entities E_m of a mention m , we choose the entity with the maximal authority score $a(v)$ as the predicted linking entity, i.e., $e_m^* = \arg \max_{v \in E_m} a(v)$.

Regarding the NIL entity problem, we use a threshold τ to determine whether we return the predicted entity e_m^* for a mention m or return NIL for all the algorithms including eigenvector centrality, traditional PageRank, PageRank with priors, traditional HITS and HITS with priors.

3.6. Experiments

We conducted extensive experiments to assess the performance of our approach using GERBIL (Usbeck et al., 2015), a general entity annotation benchmark. In this section, we firstly discuss the experimental settings and then present the evaluation results.

3. Collective Context-Aware Entity Disambiguation

	<i>ACE2004</i>	<i>AIDA/CoNLL</i>	<i>AQUAINT</i>	<i>DBpedia Spotlight</i>	<i>ITB</i>	<i>KORE30</i>	<i>MSNBC</i>	<i>N³ RSS-500</i>	<i>N³ Reuters-128</i>
Systems	Micro F1								
ADGISTIS	0.63	0.47	0.51	0.27	0.47	0.32	0.65	0.61	0.64
AIDA	0.09	0.4	0.08	0.22	0.18	0.64	0.25	0.43	0.37
Babelfy	0.52	0.54	0.68	0.53	0.37	0.74	0.64	0.45	0.45
DBpedia Spotlight	0.47	0.42	0.53	0.71	0.3	0.43	0.37	0.2	0.33
Dexter	0.52	0.4	0.52	0.29	0.21	0.2	0.35	0.37	0.36
Entityclassifier.eu	0.49	0.41	0.42	0.25	0.14	0.29	0.45	0.34	0.37
FOX	0	0.45	0	0.15	0.02	0.29	0.02	0.56	0.55
FRES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.36
FREME NER	0	0	0	0	0	0	0	0	0
KEA	0.64	0.52	0.77	0.74	0.48	0.59	0.7	0.44	0.51
NERD-ML	0.56	0.45	0.58	0.55	0.43	0.32	0.54	0.38	0.41
TagMe 2	0.67	0.47	0.71	0.67	0.37	0.57	0.57	0.47	0.43
WAT	0.64	0.58	0.72	0.66	0.41	0.59	0.62	0.44	0.51
Wikipedia Miner	0.69	0.45	0.77	0.69	0.44	0.42	0.5	0.41	0.47
NC+PRankP	0.66	0.76	0.65	0.4	0.48	0.52	0.69	0.49	0.45
NER+PRankP	0.71	0.76	0.70	0.52	0.51	0.55	0.71	0.56	0.53
N-gram+PRankP	0.65	0.78	0.8	0.51	0.52	0.51	0.57	0.63	0.54
*POS+PRankP	0.78	0.78	0.79	0.58	0.54	0.54	0.65	0.64	0.64
POS+EigenC	0.25	0.31	0.34	0.26	N/A	0.18	0.34	0.34	0.31
POS+HITS	0.17	0.28	0.11	0.33	0.07	0.43	0.16	0.4	0.22
POS+HITSP	0.68	0.69	0.62	0.44	0.47	0.5	0.63	0.59	0.56
POS+PRank	0.75	0.77	0.71	0.49	0.52	0.54	0.71	0.6	0.58

Table 3.4.: Comparison of 8 variants of our approach and 14 state-of-the-art approaches on 9 datasets using Micro F1 (best results formatted in bold), where if a system provides no results or errors, we report them as N/A (not available).

	ACE2004	AIDA/CoNLL	AQUAINT	DBpedia Spotlight	ITB	KORE50	MSNBC	N ³ RSS-500	N ³ Reuters-128
Systems	Macro F1								
ADGISTIS	0.77	0.5	0.49	0.28	0.48	0.3	0.61	0.61	0.7
AIDA	0.42	0.41	0.08	0.19	0.19	0.59	0.23	0.38	0.3
Babelfy	0.69	0.5	0.68	0.52	0.35	0.71	0.59	0.39	0.39
DBpedia Spotlight	0.67	0.44	0.51	0.69	0.28	0.39	0.36	0.17	0.26
Dexter	0.67	0.38	0.51	0.26	0.21	0.14	0.37	0.3	0.31
Entityclassifier.eu	0.66	0.41	0.38	0.2	0.16	0.26	0.44	0.32	0.34
FOX	0.37	0.44	0	0.12	0.02	0.25	0.02	0.54	0.58
FRES	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.37
FREME NER	0.37	0	0	0.02	0	0	0	0	0
KEA	0.76	0.52	0.76	0.73	0.46	0.53	0.67	0.39	0.46
NERD-ML	0.72	0.45	0.56	0.53	0.42	0.26	0.54	0.31	0.35
TagMe 2	0.78	0.46	0.69	0.66	0.36	0.49	0.57	0.39	0.36
WAT	0.76	0.59	0.72	0.67	0.39	0.48	0.57	0.37	0.43
Wikipedia Miner	0.79	0.45	0.75	0.67	0.42	0.34	0.48	0.37	0.39
NC+PRankP	0.77	0.74	0.66	0.39	0.47	0.52	0.73	0.59	0.49
NER+PRankP	0.81	0.75	0.72	0.54	0.50	0.55	0.75	0.65	0.59
N-gram+PRankP	0.8	0.79	0.8	0.47	0.51	0.5	0.62	0.7	0.63
*POS+PRankP	0.86	0.8	0.79	0.64	0.54	0.53	0.71	0.71	0.71
POS+EigenC	0.53	0.34	0.34	0.26	N/A	0.22	0.36	0.5	0.41
POS+HITS	0.61	0.37	0.12	0.34	0.07	0.43	0.22	0.55	0.42
POS+HITSP	0.8	0.73	0.62	0.42	0.46	0.49	0.66	0.66	0.66
POS+PRank	0.84	0.78	0.72	0.44	0.51	0.53	0.73	0.67	0.65

Table 3.5.: Comparison of 8 variants of our approach and 14 state-of-the-art approaches on 9 datasets using Macro F1 (best results formatted in bold), where if a system provides no results or errors, we report them as N/A (not available).

3.6.1. Experimental Setup

In the experiments, we use DBpedia 2014¹ as the knowledge base. The experiments were carried out on 9 different datasets. An overview of these datasets is shown in Table 3.3. In the following, we briefly describe these datasets and their features.

ACE2004 This dataset introduced by (Ratinov et al., 2011) is a subset of the ACE co-reference dataset, where the annotations are obtained by asking annotators on Amazon’s Mechanical Turk to link the first mention of each co-reference chain to Wikipedia.

AIDA/CoNLL This dataset introduced by (Hoffart et al., 2011) is divided into 3 chunks: Training, TestA and TestB, where only named entities are annotated. In (Hoffart et al., 2011), the first two chunks are used for training and tuning, only TestB, made up of 231 documents, is used for testing. In our experiments, we also use Training and TestA for parameter tuning and use TestB for assessing the performance of our approach.

AQUAINT This dataset introduced by (Milne & Witten, 2008b) consists of 50 newswire texts, where instead of annotating all occurrences of entities, only some important entities and their first mentions are retained to mimic the hyperlink structure in Wikipedia.

DBpedia Spotlight This dataset produced in (Mendes et al., 2011) contains quite short texts, where the mentions of both named entities and nominal entities are annotated.

IITB This dataset presented by (Kulkarni et al., 2009) contains 103 Web documents, where almost all mentions for broad types of entities including the not highly relevant ones are annotated.

KORE50 This dataset (Hoffart et al., 2011) aims for hard disambiguation tasks with very ambiguous mentions. 50 hand-crafted, difficult sentences from different domains are comprised in this dataset.

MSNBC This dataset is presented by (Cucerzan, 2007), in which all mentions of named entities are annotated in 20 news articles. It focuses on disambiguating named entities after running NER and co-reference resolution systems on newsire text.

N³ RSS-500 This dataset is one of the N³ datasets (Röder et al., 2014), where 500 sentences selected from crawled RSS feeds for a wide range of topics are annotated by domain experts.

N³ Reuters128 This is another N³ dataset (Röder et al., 2014), which contains 128 economic news articles, where the annotations of entities and mentions are determined by two domain experts.

Based on the TestA chunk and the Training chunk of the AIDA/CoNLL dataset, the parameter α in Eq. 3.3 has been tuned and we learn the threshold τ to determine whether we return the predicted entity e_m^* for a mention m as the target entity or return NIL. Regarding NER and POS based contextual entity detection, we employ Stanford Named Entity Recognizer²

¹<http://wiki.dbpedia.org/Downloads2014>

²<http://nlp.stanford.edu/software/CRF-NER.html>

and POS Tagger³. For N-gram based contextual entity detection, we extract all n-grams with $n \leq 20$.

3.6.2. Evaluation Results

We extensively evaluated various variants of our approach to entity disambiguation based on different combinations of the methods for contextual entity detection (including *NC*, *NER*, *N-gram* and *POS*, where *NC* denotes the method without using any contextual entities such that the disambiguation graph contains only candidate entities of the input mentions and the others denote the methods using NER, N-gram and POS based contextual mention detection, respectively) and the algorithms for collective entity disambiguation (including *EigenC*, *PRank*, *PRankP*, *HITS* and *HITSP*, which denote the algorithms of Eigenvector Centrality, traditional PageRank, PageRank with Priors, traditional HITS and HITS with Priors, respectively). In the experiments, we employ the measures of micro F1 and macro F1 as the quality criteria.

The experimental results show that our approach with the combination of *POS* and *PRankP*, denoted by *POS+PRankP*, achieves the best results on most datasets compared to other combinations. In the following, we focus the discussion on 8 variants of our approach with *POS* or *PRankP* involved, each of which outperforms the variants when replacing *POS* with other methods of contextual mention detection or replacing *PRankP* with other algorithms of collective entity disambiguation. We compare our approach against 14 state-of-the-art approaches using GERBIL. In addition, the impact of different solutions to contextual entity detection and collective entity disambiguation in our approach will be discussed.

Comparison with State-of-the-Art Methods

As shown in Table 3.4 and Table 3.5, we compare our approach with 14 state-of-the-art approaches on 9 datasets. The best variant of our approach *POS+PRankP* outperforms all 14 state-of-the-art approaches on 5 out of 9 datasets for both micro F1 and macro F1. Besides *POS+PRankP*, some other variants can also achieve relatively good results compared to the state-of-the-art approaches. For example, *NER+PRankP*, *N-gram+PRankP* and *POS+PRank* outperforms all state-of-the-art approaches on 4 datasets for micro F1 and on 5 datasets for macro F1, respectively.

We observe that our approach doesn't work well for two datasets, i.e., *DBpedia Spotlight* and *KORE50*, where *KEA* and *Babelify* achieve the best results for each dataset, respectively. The reason could be that the documents in these two datasets are very short and also contain very ambiguous mentions such that our approach doesn't have enough context to perform the collective entity disambiguation. For such kind of documents, the context should be extracted not only from the given document itself but also from other external resources.

³<http://nlp.stanford.edu/software/tagger.html>

3. Collective Context-Aware Entity Disambiguation

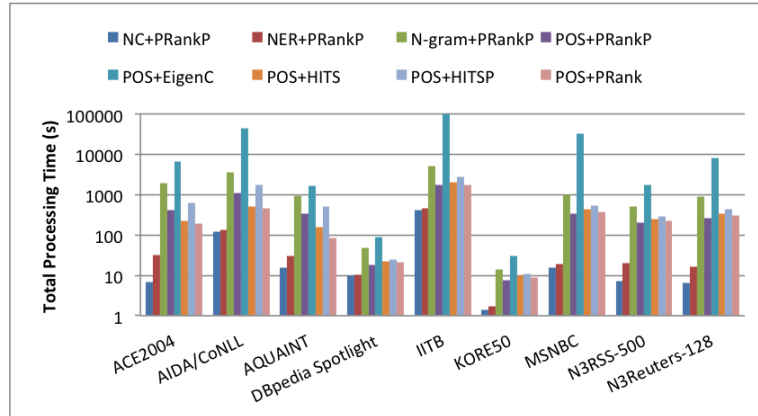


Figure 3.1.: Total processing time (s) of 8 variants of our approach.

Analysis of Contextual Entity Detection

Among the variants of our approach based on different contextual entity detection methods, *POS+PRankP* apparently achieves the best results in most cases. According to both measures of micro F1 and macro F1, it obtains the best results on 5 out of 9 datasets. Compared with *POS+PRankP*, *N-gram+PRankP* yields the best results on 2 datasets for micro F1 and 1 dataset for macro F1, and *NER+PRankP* gets the best results on 1 dataset for both micro F1 and macro F1. In general, the variants of our approach using contextual entity detection, i.e., *NER+PRankP*, *N-gram+PRankP* and *POS+PRankP*, considerably outperform *NC+PRankP* that doesn't use any contextual entities.

Note that *NC+PRankP* also achieves very good results on the *AIDA/CoNLL* and *MSNBC* datasets, where it outperforms all 14 state-of-the-art approaches. The reason could be that each document in these datasets contains quite a lot of input mentions of named entities, such that these input mentions result in more candidate entities that can be utilized by the collective disambiguation. Although the *IITB* dataset has a much higher average number of input mentions per document, many of them refer to entities that are not relevant and thus cannot be beneficial to collective entity disambiguation, such that *NC+PRankP* doesn't perform very well on *IITB*.

In addition, we investigate the impact of different methods of contextual entity detection on the runtime performance of our approach to entity disambiguation. Fig. 3.1 illustrates the total time for processing 9 datasets using different variants of our approach. We observe that *N-gram+PRankP* requires more time than *POS+PRankP*, which in turn, takes more time than *NER+PRankP* and *NC+PRankP*. This reflects the fact that *N-gram* results in more contextual entities than *POS*, which have to be taken into account by the collective disambiguation algorithms. Similarly, *POS* results in more contextual entities than *NER*. Since *NC* doesn't yield

any contextual entities, it achieves the best runtime performance.

Analysis of Collective Entity Disambiguation

We now analyze the impact of different collective entity disambiguation algorithms, where *POS* is assumed to be the method of contextual entity detection. As shown in Table 3.4 and Table 3.5, the variant using *PRankP* clearly outperforms the others. While *PRank* and *HITSP* yield relatively good results, the variants with *EigenC* and *HITS* show really poor performance.

Regarding the runtime performance as illustrated in Fig. 3.1, we observe that the variant with *EigenC* takes substantially more time, where the processing of the *IITB* dataset did not stop after running for one day such that we manually stopped it, while the variants with the other collective disambiguation algorithms exhibit only minor differences.

3.7. Related Work

In this section, we review the state-of-the-art approaches to entity disambiguation, which have been empirically compared with our approach in the experiments.

DBpedia Spotlight (Mendes et al., 2011) is one of the first approaches by combining named entity recognition and disambiguation based on DBpedia. By employing a vector space model, each entity is represented as a vector in a multidimensional word space, where term frequency (TF) and inverse document frequency (IDF) are utilized to model the relevance and importance of words. In addition, the inverse candidate frequency (ICF) is used to weight words according to their ability to distinguish between candidate entities.

Wikipedia Miner (Milne & Witten, 2008b) is one of the oldest tools widely used for entity disambiguation and linking based on Wikipedia. It provides useful statistics about anchor texts and links in Wikipedia and defines an entity relatedness measure using Wikipedia link structures. Based on a classifier using different features, e.g., prior probability, context relatedness and quality, an entity disambiguator and a link detector are provided.

NERD (van Erp et al., 2013) has been proposed for recognizing and extracting entities from tweets. Using a conditional random fields (CRF) model, entity types can be classified based on a rich feature vector composed of several linguistic features. In addition, a set of NER extractors are supported by the NERD Framework. The follow-up, NERD-ML (Rizzo et al., 2014) improved the classification task by redesigning the selection of the features.

TagMe 2 (Ferragina & Scaiella, 2012) utilizes a set of links, pages and an in-link graph from Wikipedia to annotate entities in natural language text. It first recognizes named entities by matching terms with Wikipedia anchor texts and then disambiguates the detected mentions

3. Collective Context-Aware Entity Disambiguation

using the in-link graph and page information from Wikipedia. Furthermore, the identified named entities that are considered as non-coherent to the rest of the entities in the given text are then pruned by TagMe 2.

WAT (Piccinno & Ferragina, 2014) is the successor of TagMe including a re-design of all its components, i.e., the spotter, the disambiguator and the pruner, where two sets of algorithms have been introduced: the graph-based algorithms for collective entity linking and the vote-based algorithms for local entity disambiguation, and SVM linear models are used to tune the spotter and the pruner.

AGDISTIS (Usbeck et al., 2014) is a pure entity disambiguation framework, which aims at increasing the accuracy of entity disambiguation by combining some measures for calculating string similarity, a label expansion strategy for co-referencing and the HITS algorithm for graph-based disambiguation. According to this combination, the correctness of entities detected in a given document can be significantly improved.

AIDA (Hoffart et al., 2011) only focuses on named entities and adopts the YAGO knowledge base as the entity collection to perform entity disambiguation. It relies on coherence graph building and dense subgraph algorithms, which aims at maximizing the coherence among the selected annotations.

KEA NER/NED (Steinmetz & Sack, 2013) considers heterogeneous text sources created by automated multimedia analysis as context, which have different levels of accuracy, completeness, granularity and reliability. Ambiguity is solved by selecting candidate entities with the highest probability according to the context.

Babelfy (Moro et al., 2014) is based on random walk models and a densest subgraph algorithm to tackle both word sense disambiguation and entity linking tasks in a multilingual setting depending on the BabelNet semantic network.

Dexter (Ceccarelli et al., 2013) is an open-source framework with the aim of simplifying the implementation of entity disambiguation and linking such that it allows to replace single parts of the system, where several methods have been integrated.

3.8. Conclusions

In this chapter, we presented a context-aware approach to collective entity disambiguation for the input mentions with different characteristics in a consistent manner. By leveraging the contextual entities derived from the given document and the algorithms of collective disambiguation based on Markov chains, our approach achieves promising results on various types of input mentions. Through the extensive experiments conducted on 9 different datasets, we show that our approach outperforms 14 state-of-the-art methods in most cases. The experimental results also show the limitation of our approach for short text with very ambiguous mentions. In future work, we would like to incorporate other contexts extracted from external

resources into the collective disambiguation to address the challenges of ambiguous mentions in short text.

3. *Collective Context-Aware Entity Disambiguation*

4. Salient Entity Linking

Chapter 3 focuses on the task of entity disambiguation, namely linking the entity mentions in documents with the corresponding entities in knowledge bases. In addition, for many entity-centric applications, *entity salience* for a document has become a very important factor. This raises an impending need to identify a set of salient entities that are central to the input document. In this chapter, we introduce a new task of *salient entity linking* and propose a graph-based disambiguation solution, which integrates several features, especially a topic-sensitive model based on Wikipedia categories. Experimental results show that our method significantly outperforms the state-of-the-art methods in terms of precision, recall and F1 measure.

4.1. Introduction

In recent years, large repositories of structured knowledge publicly available on the Web, such as Wikipedia, DBpedia, Freebase and YAGO, have become valuable resources for information extraction. In this regard, entity linking, which leverages such knowledge bases to link words or phrases in natural language text with the corresponding entities, has emerged as a topic of major interest.

The challenges of entity linking lie in entity recognition and disambiguation. The first stage serves to detect words or phrases in text, also called mentions, that are likely to denote entities; the second stage performs the disambiguation of the recognized mentions into entities. Many methods (Milne & Witten, 2008b; Mendes et al., 2011; Han et al., 2011; Ferragina & Scaiella, 2012; van Erp et al., 2013; Usbeck et al., 2014; Rizzo et al., 2014; Piccinno & Ferragina, 2014) have been proposed to address the problems of entity disambiguation and entity linking. However, existing methods do not take into account the importance of entities w.r.t. the topics of the input document. In this work, the relation between the candidate entities and the associated categories are utilized to opt the entities that are related to the document topics. For instance, a word “apple” appearing in a document could refer to many candidate entities in the knowledge base, such as the fruit **Apple**, its product **Apple_juice** and the technology company **Apple_Inc.**. If the word “apple” appears in an agricultural article, we tend to link it to **Apple** and **Apple_juice**, while **Apple_Inc.** is more likely to be linked when the topic of the article is about high technology.

On the other hand, *salience of entities* in a document has very practical implications in the context of entity-centric applications. For example, the major commercial Web search en-

gines have incorporated entity information from structured knowledge bases into their search results. However, many Web documents contain a lot of entities and only a small subset of entities are central to these documents, which could lead to degraded relevance of the entities extracted from the document and thus reduce the performance of entity-centric applications. Therefore, there is an impending need to identify a set of salient entities in a document that play an important role in the content of the document. In this chapter, we focus on the task of *salient entity linking*, especially the disambiguation of mentions into salient entities in a document. For this purpose, we propose a graph-based disambiguation framework utilizing a topic-sensitive model based on Wikipedia categories.

The rest of the chapter is organized as follows. We start with an overview of our framework for salient entity linking in Sec. 4.2. The details of features and measures used for salient entity disambiguation are provided in Sec. 4.3. Based on that, we discuss the graph-based disambiguation utilizing a topic-sensitive model in Sec. 4.4. Evaluation results are then presented in Sec. 4.5. Finally, we survey the related work in Sec. 4.6 and conclude in Sec. 4.7.

4.2. Overview

Before we discuss our salient entity linking framework, we first formulate the task of entity linking and then introduce the problem of salient entity linking, an extension of the general entity linking task.

Definition 2 (Entity Linking). *Let $M = \{m_1, m_2, \dots, m_p\}$ denote a set of entity mentions in a document D . Given a knowledge base KB containing a set of entities $E = \{e_1, e_2, \dots, e_n\}$, the objective of entity linking is to determine the referent entities in KB for the mentions in M , where two functions are to be found. For entity recognition, the mentions need to be extracted from D , where a recognition function $er : D \rightarrow 2^M$ will be computed. The resulting mentions (i.e., a subset $\mu \subseteq M$) are then mapped to entities in KB , where a disambiguation function¹ $ed : \mu \rightarrow E$ must be derived.*

Definition 3 (Salient Entity Linking). *Given a knowledge based KB and a document D , the recognition function of salient entity linking is same as general entity linking, i.e., $er : D \rightarrow 2^M$. For the set of mentions $\mu \subseteq M$ yielded by the recognition function, the disambiguation function $ed : \mu \rightarrow E \cup \{\text{Non-Salient}\}$, which maps the set of mentions μ to entities in the KB or to non-salient entities, must be derived, where non-salient entities, denoted by the label “Non-Salient”, are entities with no focus of attention in D , i.e., the document D is not about such entities.*

¹If out-of-knowledge-base entities are supported, we have the disambiguation function $ed : \mu \rightarrow E \cup \{NIL\}$, which maps a set of mentions to entities in KB or to NIL .

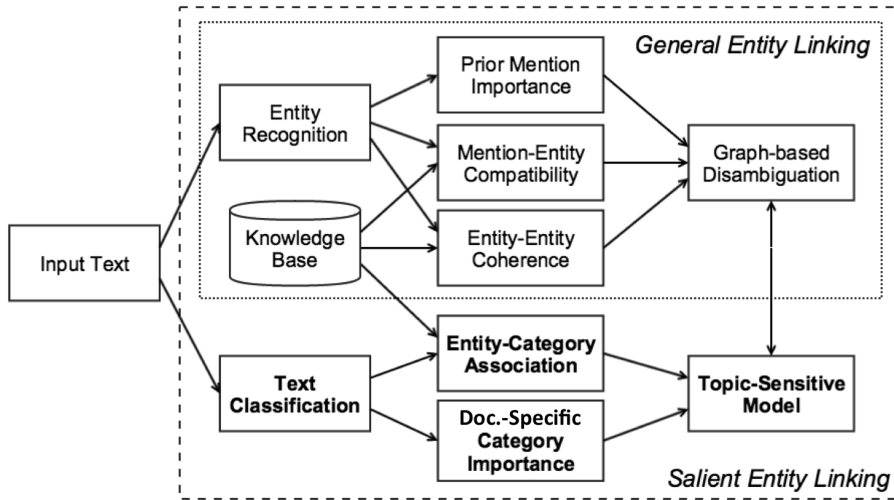


Figure 4.1.: Salient entity linking framework.

In order to address the problem of salient entity linking for an input document, we propose a framework consisting of several components as shown in Fig. 4.1. In the following, we briefly introduce the components w.r.t. general entity linking and the extensions for salient entity linking by utilizing a topic-sensitive model. The details about the features as well as the graph model and algorithm will be discussed in Sec. 4.3 and Sec. 4.4, respectively.

For both general and salient entity linking, the input text is first processed by a method for *entity recognition*, e.g., the Stanford NER Tagger (Finkel et al., 2005) that tags only named entities and our recent work (Zhang et al., 2015a) that aims to detect both named entities and nominal entities. Entity recognition detects the boundaries of mentions without knowing the actual referred entities or whether they are salient or non-salient entities. Then, these detected entity mentions serve as the input of entity disambiguation in the next step.

Regarding entity disambiguation for *general entity linking*, we use a method similar to the work discussed in Chapter 3, which is briefly described in the following. Given a detected mention, its candidate referent entities are firstly extracted from the knowledge base using a dictionary containing all entity and their surface forms. Then, our framework combines different features including *prior mention importance*, *mention-entity compatibility* and *entity-entity coherence*. The feature of *prior mention importance* assigns the prior importance to each detected mention as weight and it will be used as the initial evidence for graph-based disambiguation. While the local feature of *mention-entity compatibility* captures the most likely entity behind the mention and the entity that best fits the context, the global feature of *entity-entity coherence* collectively captures the linked entities in a document that are related to each other. These features are then employed by *graph-based disambiguation* based on a

personalized PageRank algorithm, which has been shown to be very effective in Chapter 3.

Concerning entity disambiguation for *salient entity linking*, the main difference from the general task is that for each mention the goal is to output not only the actual referent entity in the knowledge base but also the *non-salient* label in case that the corresponding entity is not salient w.r.t. the input document. To aim for salient entity linking, we first perform *text classification* on the input text using a multi-class support vector machine (SVM) classifier based on 16 Wikipedia categories² aligned with the training corpus. For each category, we compute the category probability of the input document that serves as the feature of *document-specific category importance*. In addition, we compute the strength of *entity-category association* based on the depth between each candidate entity and its categories. Such features are then incorporated into the *graph-based disambiguation* using a topic-sensitive PageRank algorithm, an extension of the algorithm for general entity linking with category information.

4.3. Features and Measures

In this section, we discuss the features and measures needed for both general and salient entity linking, while the graph model and algorithm will be presented in Sec. 4.4.

4.3.1. General Features

The features and measures for general entity linking are similar to the ones used for entity disambiguation described in Chapter 3. In order to make this chapter self-contained, some of the material in the following is repeated from Sec. 3.4 of Chapter 3.

Prior Mention Importance

For determining the *prior mention importance*, different measures have been used in the literature, such as the TFIDF score (Han et al., 2011). Instead, we employ the Wikipedia link structures. As each Wikipedia article describes an entity, article titles, redirect pages and link anchors can be used to refer to the entity. Based on the above sources, we extract all surface forms of entities. In addition, we define the probability $P(s)$ that captures how likely a surface form s is an entity mention in a document as

$$P(s) = \frac{\text{count}_{\text{anchor}}(s)}{\text{count}_{\text{anchor}}(s) + \text{count}_{\text{raw}}(s)} \quad (4.1)$$

²In this work, we employ the 16 second-level categories including *Mathematics, People, Science, Sport, Geography, Culture, Politics, Nature, Technology, Education, Health, Business, Belief, Society, Life and Concepts* in Wikipedia, where the first-level category is the fundamental category.

where $count_{anchor}(s)$ denotes the number of articles that contain s as anchor text and $count_{raw}(s)$ denotes the number of articles where s appears as raw text.

Mention-Entity Compatibility

For each entity mention m , we calculate the semantic similarity $SS(m, e)$ representing the local *mention-entity compatibility* of m and its referent entity e as follows

$$SS(m, e) = \alpha \cdot LP(m, e) + (1 - \alpha) \cdot CS(m, e) \quad (4.2)$$

where $LP(m, e)$ is the link probability of e for m and $CS(m, e)$ is the context similarity between m and e , α is a tunable parameter. An entity e in KB is characterized by its textual description $e.c$, called *context* of e and a mention m is characterized by its name $m.s$ as a surface form of the corresponding entity and its local surrounding sentences $m.c$, called *context* of m . The link probability $LP(m, e)$ can be calculated using the probability $P(e|s)$ capturing how likely the surface form s refers to the entity e as follows

$$LP(m, e) = P(e|m.s) = \frac{count_{link}(e, m.s)}{\sum_{e_i \in E_s} count_{link}(e_i, m.s)} \quad (4.3)$$

where $count_{link}(e, s)$ denotes the number of links using s as anchor text pointing to e as destination and E_s is the set of entities that have the surface form s . The context similarity $CS(m, e)$ between m and e can be calculated using cosine similarity on the term vector of the context of m and e as

$$CS(m, e) = \cos(e.c, m.c) = \frac{\langle e.c, m.c \rangle}{|e.c| \cdot |m.c|} \quad (4.4)$$

Entity-Entity Coherence

The disambiguation process is based on the feature of *entity-entity coherence*, which collectively captures the referent entities of the mentions contained in the same document that are related to each other. In this regard, we calculate the semantic relatedness between each pair of entities e_i and e_j by adopting the Wikipedia link-based measure described in (Milne & Witten, 2008a; Milne & Witten, 2008b), which is originally modeled after the Normalized Google Distance (NGD) (Cilibrasi & Vitányi, 2007), as follows

$$SR(e_i, e_j) = 1 - \frac{\log(\max(|E_i|, |E_j|)) - \log(|E_i \cap E_j|)}{\log(|E|) - \log(\min(|E_i|, |E_j|))} \quad (4.5)$$

where E_i and E_j are the sets of entities that link to e_i and e_j in KB respectively, and E is the set of all entities in KB.

4.3.2. Salient Features

As shown in Chapter 3, the general features discussed above can actually help with entity disambiguation for general entity linking. However, such features do not reflect any information of entity salience so that it is not sufficient to only use them for salient entity linking. In this regard, we introduce two additional features in the following.

Document-specific Category Importance

For text classification of the input document, we employ John C. Platt’s sequential minimal optimization algorithm for training a SVM classifier (Platt, 1999; Keerthi et al., 2001). Multi-category problems are solved using pairwise classification. To obtain proper probability estimates, we use the option that fits logistic regression models to the outputs of the SVM. In our multi-category scenario, the predicted probabilities are coupled using Hastie and Tibshirani’s pairwise coupling method (Hastie & Tibshirani, 1997). All these algorithms have been integrated into Weka³, a collection of machine learning algorithms for data mining tasks. Based on that, we calculate the category probability $P(c_i)$ of the input text for each assigned category c_i , which reflects the *document-specific category importance*.

Entity-Category Association

The candidate referent entities are connected to the Wikipedia categories resulting from text classification on the input text, if there exists a path between the corresponding entities and categories in Wikipedia. This can be done by performing a breadth-first search starting from the fundamental category that forms the root of Wikipedia’s category hierarchy to each entity. In order to measure the *entity-category association* between an entity e and its assigned category c , we define the distance $d(e, c)$ as the minimum depth at which the entity e is located in Wikipedia’s category tree with the category c as the root. Then we calculate the semantic association $SA(e, c)$ between entity e and category c as follows

$$SA(e, c) = \frac{1}{d(e, c)} \quad (4.6)$$

4.4. Graph Model and Algorithm

Based on the features and measures discussed in Sec. 4.3, we construct a directed weighted graph $G = \{N, R\}$, called *disambiguation graph*, where $N = N_M \uplus N_E \uplus N_C$ is the disjoint union of *mention* nodes N_M , *entity* nodes N_E and *category* nodes N_C , and R is the set of

³<http://www.cs.waikato.ac.nz/ml/weka>

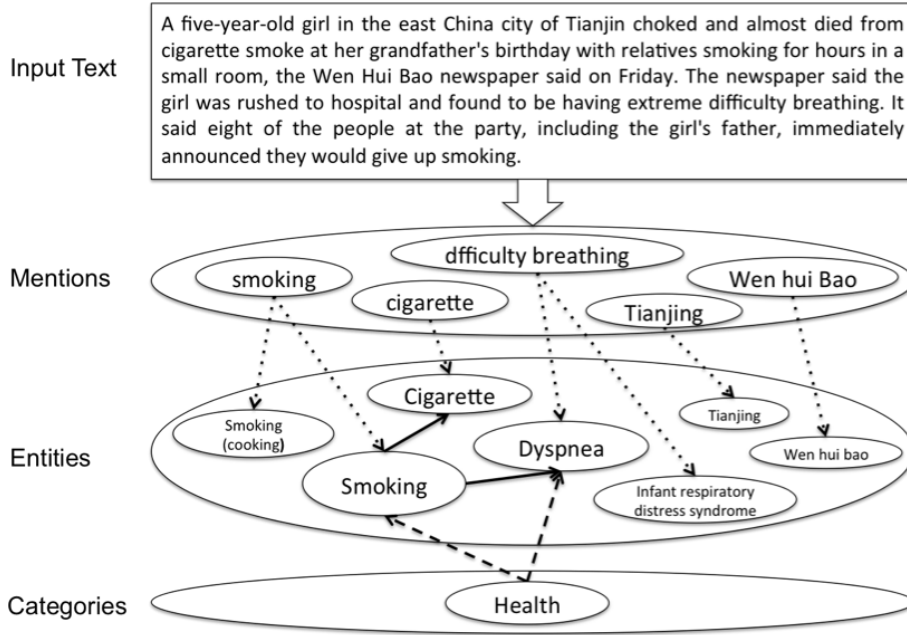


Figure 4.2.: Example of graph-based disambiguation utilizing a topic-sensitive model.

directed edges representing relationships between nodes. All detected mentions and their candidate referent entities are added into N_M and N_E , respectively, while the categories that the input text belongs to are added into N_C . For each mention m and its candidate entity e , we add an edge from m to e into R . Additionally, we add an edge between e_i and e_j into R if they are connected in KB. Furthermore, for each association between an entity e and a category c , an edge from c to e will be added into R . An example of the disambiguation graph is illustrated in Fig. 4.2.

Once the disambiguation graph G is built, we apply the topic-sensitive PageRank algorithm, which has been proposed in (Haveliwala, 2002; Haveliwala, 2003) for improving the ranking of Web search results by using the categories of query keywords. Different from the PageRank algorithm used in Chapter 3, two types of new elements are included in the disambiguation graph G , i.e., category nodes as well as edges between entity nodes and category nodes. The calculation of the PageRank vector Pr over G is equivalent to resolving the following equation

$$Pr = d \cdot T \cdot Pr + (1 - d) \cdot v \quad (4.7)$$

where T is the transition probability matrix, v is the initial evidence vector and d is the so called damping factor, usually set as 0.85. Each entry T_{ij} in T is the evidence propagation

4. Salient Entity Linking

ratio from node i to node j , which is computed in Eq. 4.8.

$$T_{ij} = \begin{cases} \frac{SS(m_i, e_j)}{\sum_{k \in N_E(i)} SS(m_i, e_k)} & \text{if } i \in N_M, j \in N_E \\ \frac{SR(e_i, e_j)}{\sum_{k \in N_E(i)} SR(e_i, e_k)} & \text{if } i \in N_E, j \in N_E \\ \frac{SA(c_i, e_j)}{\sum_{k \in N_E(i)} SA(c_i, e_k)} & \text{if } i \in N_C, j \in N_E \end{cases} \quad (4.8)$$

where $N_E(i)$ is the set of entity nodes such that for each node $k \in N_E(i)$, there is an edge from i to k in G . The entry v_i in v is the initial evidence representing the prior importance of a mention m_i if $i \in N_M$ or the document-specific importance of an category c_i if $i \in N_C$, which is calculated as follows

$$v_i = \begin{cases} \frac{\lambda \cdot P(m_i)}{\lambda \cdot \sum_{k \in N_M} P(m_k) + (1-\lambda) \cdot \sum_{k \in N_C} P(c_k)} & \text{if } i \in N_M \\ \frac{(1-\lambda) \cdot P(c_i)}{\lambda \cdot \sum_{k \in N_M} P(m_k) + (1-\lambda) \cdot \sum_{k \in N_C} P(c_k)} & \text{if } i \in N_C \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

where λ is a tunable parameter, which reflects the sensitivity of prior mention importance and document-specific category importance to the final probability of each candidate entity. When $\lambda = 1$, our method reduces to general entity linking without considering the topic-sensitive model. In contrast, when $\lambda = 0$, the initial evidence of the graph-based disambiguation only depends on the category importance.

As a result of the topic-sensitive PageRank algorithm, each candidate entity e receives a final probability $P(e)$. For each mention m having a set of candidate entities E_m , we choose the entity with the maximal probability as the predicted linking entity, i.e., $e_m = \arg \max_{e \in E_m} P(e)$. The process discussed above doesn't distinguish between salient and non-salient entities. In order to deal with salient entity linking, one important task of the *topic-sensitive model* is to validate whether the predicted linking entity e_m for mention m is a salient entity. For this purpose, we learn a threshold τ such that if $P(e_m)$ is greater than τ we return e_m as the linking entity for m , otherwise we return the label *Non-Salient*.

4.5. Experiments

We now discuss the experiments we conducted to assess the performance of our approach to salient entity linking. In this section, we firstly discuss the experimental settings and then present the evaluation results.

Corpus	#Doc.	#Words	#Mentions	#Salient Entities	#Non-Salient Entities
Reuters-128	128	33,413	4,429	2,554 (58%)	1,875 (42%)

Table 4.1.: Statistics of Reuters-128 entity salience dataset.

4.5.1. Experimental Setup

In our experiments, we use DBpedia 2014⁴ as the knowledge base and the Wikipedia snapshot of July 2013 as the auxiliary data source. The experiments were carried out on the Reuters-128 entity salience dataset⁵, which is an extension of a part of the N3 entity linking datasets (Röder et al., 2014). The Reuters-128 dataset is an English corpus and it contains 128 economic news articles. The dataset contains information for 880 named entities with their position in the document and a URI of a DBpedia resource identifying each entity. The salience dataset extends the Reuters-128 dataset also with 3,551 common entities. Some statistics about the Reuters-128 salience dataset can be found in Table 4.1.

In order to construct the dataset, entity salience information was obtained by crowdsourcing salience information using the CrowdFlower platform. For each named and common entity in the Reuters-128 dataset, the authors of the dataset collected at least three judgements. Only judgments from annotators with trust score higher than 70% were considered as trusted judgements. If the trust score of an annotator falls below 70%, all his/her judgements were disregarded. Finally, each named and common entity in the dataset has been classified in one of the following classes⁶:

- *Most Salient* - Entities with the highest focus of attention in the article. The document is mostly about these entities, or the entities play a prominent role in the content of the article.
- *Less Salient* - Entities with less focus of attention in the article. The entities play an important role in some parts of the content of the article.
- *Not Salient* - The article is really not about the entities

4.5.2. Evaluation Results

In our experiments, we consider the entities in both classes *Most Salient* and *Less Salient* as salient entities, while entities belonging to *Not Salient* are considered as non-salient entities. Using the Reuters-128 entity salience dataset, we conducted the experiments to compare our approach with several existing solutions to entity linking, i.e., DBpedia Spotlight (Mendes

⁴<http://wiki.dbpedia.org/Downloads2014>

⁵<https://github.com/KIZI/ner-eval-collection>

⁶<http://ner.vse.cz/datasets/entitysalience-collection>

4. Salient Entity Linking

Methods	Mic. Prec.	Mic. Rec.	Mic. F1	Mac. Prec.	Mac. Rec.	Mac. F1.
DBpedia Spotlight	0.45	0.39	0.41	0.45	0.37	0.40
Wikipedia Miner	0.60	0.48	0.54	0.60	0.42	0.52
NERD-ML	0.67	0.50	0.57	0.65	0.46	0.54
WAT	0.35	0.32	0.34	0.36	0.33	0.34
AGDISTIS	0.73	0.50	0.59	0.73	0.48	0.58
Our Method (General)	0.70	0.46	0.56	0.69	0.45	0.55
Our Method (Salient)	0.83	0.51	0.63	0.82	0.50	0.62

Table 4.2.: The experimental results of salient entity linking.

et al., 2011), Wikipedia Miner (Milne & Witten, 2008b), NERD-ML (van Erp et al., 2013; Rizzo et al., 2014), WAT (Ferragina & Scaiella, 2012; Piccinno & Ferragina, 2014), AGDISTIS (Usbeck et al., 2014). In addition, we evaluated two variants of our approach, one employs only the graph-based disambiguation for general entity linking (i.e., $\lambda = 1$) and the other integrates the topic-sensitive model with the goal of salient entity linking (i.e., $\lambda = 0.2$). All the methods should label each mention with either the actual referent entity or non-salient entity. Note that we restrict the input to the labeled mentions to compare the method’s ability to distinguish between salient entity and non-salient entity, not its ability to recognize entity names in the input text. The adopted evaluation criteria include Micro-Precision, Micro-Recall, Micro-F1, Macro-Precision, Macro-Recall and Macro-F1.

The experimental results are shown in Table 4.2. By utilizing the topic-sensitive model, our approach to salient entity disambiguation significantly outperforms the baselines in terms of all evaluation criteria. Compared with the best results achieved by other solutions, our method of salient entity linking can produce the improvements of 14% for Micro-Precision, 2% for Micro-Recall, 7% for Micro-F1, 12% for Macro-Precision, 4% for Macro-Recall, 7% for Macro-F1. Regarding the two variants of our approach, it clearly shows that the topic-sensitive model indeed contributes to the final performance improvement, i.e., 19% for Micro-Precision, 11% for Micro-Recall, 13% for Micro-F1, 19% for Macro-Precision, 11% for Macro-Recall, 13% for Macro-F1.

4.6. Related Work

Entity linking and disambiguation from natural language text to knowledge bases have been extensively investigated in many fields of research, such as natural language processing (NLP), information retrieval (IR), knowledge extraction and Semantic Web. We have already introduced many state-of-the-art approaches in Sec. 3.7 of Chapter 3, which will be skipped here.

Another related area that has been well studied is the understanding of document aboutness. From a practical perspective, there are many methods that have decomposed aboutness

and focused on detecting different aspects of aboutness such as key terms (Yih et al., 2006; Irmak et al., 2009; Paranjpe, 2009). As one of the state-of-the-art work, the method proposed in (Paranjpe, 2009) focuses on the detection of key terms that best reflect the central topics of a document. User feedback available in click logs of a Web search engine has been used as training data for a supervised salience scoring function. In this way, the notion of document aboutness is considered as the identification of salient terms.

Motivated by the real demands of entity-centric applications on the Web, the recent work in (Gamon et al., 2013) considers salient entities as an important aspect of document aboutness. The authors firstly use the named entity recognition (NER) system as a candidate entity generator. Then, they develop a supervised model for learning document aboutness through identification of salient entities from the set of candidate entities. This solution can be seen as related to the methods based on key terms, where the recognized named entities, i.e., the words and phrases as entity mentions, are a subset of terms in a document.

Different from the previous work on entity salience, our proposed method considers the canonical entities (including both named entities and nominal entities) in knowledge bases instead of only the words or phrases as mentions of named entities detected by NER. On the other hand, our work, in contrast to the existing work on entity linking, does not aim at identification of all entities in knowledge bases, but only the most central ones.

4.7. Conclusions

In this chapter, we introduce the task of salient entity linking that existing entity linking solutions cannot well address. For tackling this new problem, we propose a graph-based disambiguation framework, which integrates several features including prior mention importance, mention-entity compatibility, entity-entity coherence and in particular a topic-sensitive model capturing entity-category association and document-specific category importance. We have experimentally shown that our approach to salient entity linking achieves a significant improvement over the existing solutions. The evaluation results also show that the incorporation of the topic-sensitive model indeed helps with the salient entity disambiguation.

4. *Salient Entity Linking*

Part III.

Semantic Search

5. Time-Aware Entity Recommendation

There has been an increasing effort to develop techniques for related entity recommendation, where the task is to retrieve a ranked list of related entities given a keyword query. Another trend in the area of information retrieval (IR) is to take temporal aspects of a given query into account when assessing the relevance of documents. However, while this has become an established functionality in document search engines, the significance of time has not yet been recognized for entity recommendation. In this chapter, we address this gap by introducing the task of *time-aware entity recommendation*. We propose the first probabilistic model that takes time-awareness into consideration for entity recommendation by leveraging heterogeneous knowledge of entities extracted from different data sources publicly available on the Web. We extensively evaluate the proposed approach and our experimental results show considerable improvements compared to time-agnostic entity recommendation approaches.

5.1. Introduction

In recent years, many research activities involving entities have emerged and increasing attention has been devoted to technologies aimed at identifying entities related to a user's information need. *Entity search* has been defined as finding an entity in the knowledge base that is explicitly named in a keyword query (Pound et al., 2010). A variant of entity search is *related entity recommendation*, where the goal is to rank relationships between a query entity and other entities in a knowledge base (van Zwol et al., 2010; Kang et al., 2011). In the context of Web search, *entity recommendation* has been defined as finding the entities related to the entity appearing in a Web search query (Blanco et al., 2013).

On the other hand, temporal dynamics and their impact on information retrieval (IR) have drawn increasing attention in the last decade. In particular, the study of document relevance by taking into account the temporal aspects of a given query is addressed within *temporal IR* (Kanhabua et al., 2015). To support a temporal search, a basic solution is to extend keyword search with the creation or publication date of documents, such that search results are restricted to documents from a particular time period given by a time constraint (Nørsvåg, 2004; Berberich et al., 2007). This feature is already available in every major search engine, e.g., Google also allows users to search Web documents using a keyword query and a customized time range. For the effectiveness of temporal IR, the time dimension has been incorporated

5. Time-Aware Entity Recommendation

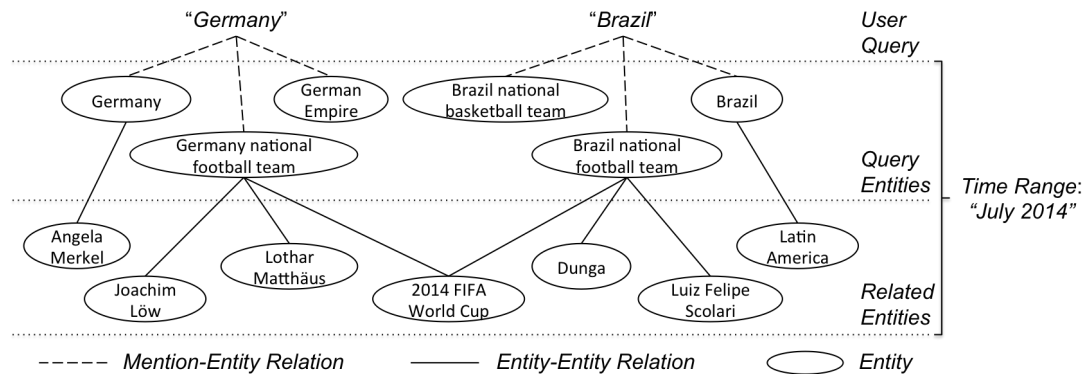


Figure 5.1.: Examples of the candidate query entities and related entities for the user query “Germany Brazil” and the given time range “July 2014”.

into retrieval and ranking models, also called *time-aware retrieval and ranking*. More precisely, documents are ranked according to both textual and temporal similarity w.r.t. the given temporal information needs (Kanhabua et al., 2015).

Inspired by temporal IR, we believe that the time dimension could also have a strong influence on entity recommendation. Existing entity recommendation systems aim to link the initial user query to its related entities in the knowledge base and provide a ranking of them. Typically, this has been done by exploiting the relationships between entities in the knowledge base (van Zwol et al., 2010; Kang et al., 2011; Blanco et al., 2013). However, the (temporal) entity importance and relatedness is often significantly impacted by real-world events of interest to users. For example, a sports tournament could drive searches towards the teams and players that participate in the tournament and the acquisition of a company by another company could establish a new relationship between them and thus affect their relatedness. Some efforts have already been devoted to improve the quality of recommendations in particular with respect to data freshness. For example, Sundog (Fischer et al., 2015) uses a stream processing framework for ingesting large quantities of Web search log data at high rates such that it can compute feature values and entity rankings in much less time compared to previous systems, such as Spark (Blanco et al., 2013), and thus can use more recently collected data for the ranking process. However, the time-awareness, which should be a crucial factor in entity recommendation, has still not been addressed.

Let us suppose users issue the keyword query “Germany Brazil” (see Fig. 5.1). Then they are likely looking for related geographic or political entities. However, when additionally specifying the time range “July 2014”, their interest is more likely related to the German and Brazilian national football teams during the 2014 FIFA World Cup. Obviously, once time information is available, the goal for a related entity search approach should be to improve entity recommendation such that the ranking of related entities depends not only on entity information in the knowledge base but also on the real-world events taking place in a specific time period. Therefore, it is essential to make *time-awareness* a top priority in entity

recommendation when a customized time range is given.

We introduce the problem of *time-aware entity recommendation* (TER), which allows users to restrict their interests of entities to a customized time range. In general, the goal of TER is to (1) *disambiguate the query entities* mentioned in the user query and (2) *find the related entities* to the query entities as well as (3) *rank all these query entities and related entities according to time* in order to match information needs of users, where the time dimension plays an important role. As shown in Fig. 5.1, the keywords “Germany” and “Brazil” result in different potential query entities. Since `Germany_national_football_team` and `Brazil_national_football_team` are of particular interest during the given time range “July 2014”, they should more likely be the intended query entities. For each query entity, its related entities will be found through the relations between entities, which can also be influenced by the time dimension. For example, the query entity `Brazil_national_football_team` results in the related entities `Dunga`, the current coach of Brazilian national football team, and `Luiz_Felipe_Scolari`, the coach during 2014 FIFA World Cup. By taking into account the time dimension, `Luiz_Felipe_Scolari` should be preferred over `Dunga` since the user requests information from July 2014.

To achieve this, we propose a probabilistic model by decomposing the TER task into several distributions, which reflect heterogeneous entity knowledge including *popularity*, *temporality*, *relatedness*, *mention* and *context*. The parameters of these distributions are then estimated using different real-world data sources, namely Wikipedia¹, Wikilinks², Wikipedia page view statistics³ and a multilingual real-time stream of annotated Web documents. Please note that the data sources used by existing systems are mostly not publicly accessible. Particularly the major Web search engines keep their own usage data, like query terms and search sessions as well as user click logs and entity pane logs, secret, since they are crucial to optimizing their own entity recommendation systems, like the ones of Yahoo! (Blanco et al., 2013; Fischer et al., 2015) and Microsoft (Yu et al., 2014; Bi et al., 2015). In contrast, our approach does not rely on datasets taken from commercial Web search engines, but only resorts to data sources publicly available on the Web.

The main contributions are: (1) We introduce a formal definition of the TER problem (2) and propose a statistically sound *probabilistic model* that incorporates *heterogeneous entity knowledge* including the *temporal context*. (3) We show how all parameters of our model can be effectively estimated solely based on data sources *publicly available* on the Web. (4) Due to the lack of benchmark datasets for the TER challenge, we have created *new datasets* to enable empirical evaluations and (5) the results show that our approach improves the performance considerably compared to time-agnostic approaches.

The rest of the chapter is organized as follows. We present the overall approach, especially

¹<https://dumps.wikimedia.org/>

²<http://www.iesl.cs.umass.edu/data/wiki-links/>

³<https://dumps.wikimedia.org/other/pagecounts-raw/>

the probabilistic model in Sec. 5.2. Then, we describe the estimation of model parameters in Sec. 5.3. The experimental results are discussed in Sec. 5.4. Finally, we survey the related work in Sec. 5.5 and conclude in Sec. 5.6.

5.2. Overview

We first formally define the *time-aware entity recommendation* (TER) task and then describe the probabilistic model of our approach.

Definition 4 (Time-Aware Entity Recommendation). *Given a knowledge base with a set of entities $E = \{e_1, \dots, e_N\}$, the input is a keyword query q , which refers to one or more entities, and a continuous date range $t = \{d_{start}, \dots, d_{end}\}$ where $d_{start} \leq d_{end}$, and the output is a ranked list of entities that are related to q , especially within t .*

To facilitate the discussion in the following, we first introduce the concepts of *mention* and *context*. For a keyword query q , a *mention* is a term in q that refers to an entity e_q , also called *query entity*, and the *context* of e_q is the set of all other mentions in q except the one for e_q . For each query entity e_q , the keyword query q can be decomposed into the mention and context of e_q , denoted by s_{e_q} and c_{e_q} respectively. For example, given the query entity `Germany_national_football_team`, the keyword query “*Germany Brazil*” results in the mention “*Germany*” and the context {“*Brazil*”}.

In this work, we use DBpedia as the knowledge base, which contains an enormous number of entities in different domains by extracting various kinds of structured information from Wikipedia, where each entity is tied to a Wikipedia article.

5.2.1. Probabilistic Model

We formalize the TER task as estimating the probability $P(e|q, t)$ of each entity e given a keyword query q and a date range t . The goal is then to find a ranked list of top- k entities e , which maximize the probability $P(e|q, t)$. Based on Bayes’ theorem, the probability $P(e|q, t)$ can be rewritten as follows

$$P(e|q, t) = \frac{P(e, q, t)}{P(q, t)} \propto P(e, q, t) \quad (5.1)$$

where the denominator $P(q, t)$ can be ignored as it does not influence the ranking. The joint probability $P(e, q, t)$ is then given as

$$\begin{aligned} P(e, q, t) &= \sum_{e_q} P(e_q, e, q, t) = \sum_{e_q} P(e_q, e, s_{e_q}, c_{e_q}, t) \\ &= \sum_{e_q} P(e)P(t|e)P(e_q|e, t)P(s_{e_q}|e_q, e, t)P(c_{e_q}|e_q, e, t) \end{aligned} \quad (5.2)$$

$$= \sum_{e_q} P(e)P(t|e)P(e_q|e, t)P(s_{e_q}|e_q)P(c_{e_q}|e_q, t) \quad (5.3)$$

where we assume in (5.2) s_{e_q} and c_{e_q} are conditionally independent given e_q and t , in (5.3) s_{e_q} is conditionally independent of e and t given e_q , and c_{e_q} is conditionally independent of e given e_q and t . The intuition behind these assumptions is that a mention s_{e_q} should only rely on the query entity e_q it refers to and a context c_{e_q} that appears together with e_q should depend on both e_q and t .

The main problem is then to estimate the components of $P(e, q, t)$ including the *popularity* model $P(e)$, the *temporality* model $P(t|e)$, the *relatedness* model $P(e_q|e, t)$, the *mention* model $P(s_{e_q}|e_q)$ and the *context* model $P(c_{e_q}|e_q, t)$.

5.2.2. Data Sources

To derive the estimation of these distributions in our model, we present several publicly available data sources. Based on these data sources, we discuss the details of model parameter estimation in Sec. 5.3.

Wikipedia and Wikilinks. Wikipedia provides several resources, including article titles, redirect pages and anchor text of hyperlinks, that associate each entity with terms referring to it, also called *surface forms* (Shen et al., 2012). Wikilinks (Singh et al., 2012) also provides surface forms of entities by finding hyperlinks to Wikipedia from a Web crawl and using anchor text as mentions. Based on such sources, we construct a dictionary that maps each surface form to the corresponding entities.

Based on the observation that a more popular entity usually has more pages linking to it, we take link frequency as an indicator of *popularity*. For example, in Wikipedia the famous basketball player Michael Jeffrey Jordan is linked over 10 times more than the Berkeley professor Michael I. Jordan.

Wikipedia link structure has also been used to model *entity relatedness* (Milne & Witten, 2008a), without considering temporal aspects, where the intuition is that Wikipedia pages containing links to both of the given entities indicate relatedness, while pages with links to only one of the given entities suggest the opposite.

Page View Stream. Wikipedia page view stream provides the number of times a particular Wikipedia page is requested per hour and thus can be treated as a query log of entities. In general, a well-known entity usually gets more page views than the obscure ones, such that the page view frequency also captures the *popularity* of entities.

In addition, an entity is likely to get more page views when an event related to it takes place. For example, during the FIFA World Cup, many participating football teams and players will get more page views. This explains the significant page view spike during an event when the entity receives media coverage, which has been utilized for the event detection task (Ciglan & Nørnvåg, 2010). In this sense, the page view spike captures a user-driven measure of the *temporality* of entities.

Furthermore, an event could result in more page views for all the involved entities. For example, when Facebook acquires WhatsApp, both of them get high page view spikes. Based on this observation, simultaneous page view spikes of entities can help with modeling *the dynamic relatedness* between entities.

Annotated Web Document Stream. Another data source is a real-time aggregated stream of semantically annotated Web documents. We first employ a *news feed aggregator*⁴ to acquire a multilingual real-time stream of news articles publicly available on the Web (Trampuš & Novak, 2012), where the enormous number of collected Web documents are in various languages, such as English (50% of all articles), German (10%), Spanish (8%) and Chinese (5%). Then we employ a cross-lingual semantic annotation system⁵ to annotate the multilingual Web documents with DBpedia entities, i.e., to link entity mentions to their referent entities (Zhang & Rettinger, 2014). Based on that, entity co-occurrence statistics extracted from the annotated Web documents can help to identify dynamically related entities and the co-occurrence frequency can be utilized to measure the *dynamic relatedness* between entities w.r.t. a specific time range.

5.2.3. Candidate Selection

As there are millions of entities in DBpedia, it is extremely time-consuming to calculate $P(e, q, t)$ for all entities. To improve the efficiency of TER, we employ a candidate selection process to filter out the impossible candidates. Given a query q and a date range t , the candidate related entities are generated in three different ways: (1) Based on the dictionary containing entities and their surface forms extracted from Wikipedia and Wikilinks datasets, all query entities, whose mentions can be found in q , are selected as a set of candidates, denoted by E_q . (2) Given the set of subject, predicate and object triples $\{(s, p, o)\}$ in DBpedia, where all subjects and objects are entities, the potential candidate related entities that have a relation to the query entities are identified as $\{e|\exists p : (e, p, e_q), e_q \in E_q, e \in E\} \cup \{e|\exists p :$

⁴http://newsfeed.ijs.si/visual_demo/

⁵<http://km.aifb.kit.edu/sites/xlisa/>

$(e_q, p, e), e_q \in E_q, e \in E\}$. (3) By analyzing the annotated Web documents, the entities that co-occur with any query entities $e_q \in E_q$ in the Web documents published during the date range t are also considered as candidate related entities.

5.3. Model Parameter Estimation

Our probabilistic model is parameterized by $\Phi_e = P(e)$, $\Phi_{t|e} = P(t|e)$, $\Phi_{e'|e,t} = P(e'|e, t)$, $\Phi_{s|e} = P(s|e)$ and $\Phi_{c|e,t} = P(c|e, t)$. In the following, we present the details of parameter estimation based on the introduced data sources.

5.3.1. Popularity Model

The distribution $P(e)$ captures the popularity of entity e . By leveraging both Wikipedia link structure and page view statistics, we first calculate $C(e)$ as

$$C(e) = C_{link}(e) + \beta C_{view}(e) \quad (5.4)$$

where $C_{link}(e)$ denotes the number of links pointing to e and $C_{view}(e)$ denotes the average number of page views on e per day. While $C_{link}(e)$ represents the prior popularity of e in Wikipedia, $C_{view}(e)$ captures the popularity of e based on user interests. Due to the different scales of link and page view frequencies, $C_{view}(e)$ is adjusted by a balance parameter $\beta = \frac{\text{total number of links in Wikipedia}}{\text{average number of page views per day}}$, which accounts for the difference in frequencies of Wikipedia links and per-day page views. Then the probability $P(e)$ is estimated as follows

$$P(e) = \frac{\log(C(e)) + 1}{\sum_{e_i \in W} \log(C(e_i)) + |W|} \quad (5.5)$$

where W denotes the set of all entities. The estimation is smoothed using Laplace smoothing for avoiding the zero probability problem.

5.3.2. Temporality Model

The distribution $P(t|e)$ captures the temporality of entity e w.r.t. date range t . We employ the page view statistics as a proxy for interest of each entity and equate the page view spike with it. For each entity e , we track its per-day page view counts for each date d . Then we compute the mean $\mu(e, d)$ and standard deviation $\sigma(e, d)$ of page views for entity e in a window of n days before d

$$\mu(e, d) = \frac{1}{n} \sum_{d_i=d-n}^{d-1} C(e, d_i) \quad (5.6)$$

$$\sigma(e, d) = \sqrt{\frac{1}{n} \sum_{d_i=d-n}^{d-1} (C(e, d_i) - \mu(e, d))^2} \quad (5.7)$$

where $C(e, d_i)$ denotes the number of page views of e on date d_i . Inspired by the work in (Osborne et al.,), we calculate the page view spike $S(e, d)$ of entity e on date d as

$$S(e, d) = \begin{cases} \frac{C(e,d)-\mu(e,d)}{\sigma(e,d)} & \text{if } \frac{C(e,d)-\mu(e,d)}{\sigma(e,d)} \geq \kappa, \\ 0 & \text{otherwise} \end{cases} \quad (5.8)$$

where we assume that only the page view count $C(e, d)$ that is abnormally large compared with the previously seen page views of e , i.e. $\frac{C(e,d)-\mu(e,d)}{\sigma(e,d)} > \kappa$ (κ is a fixed parameter set as 2.5 here), indicates an event and thus will be taken into account to compute the page view spike $S(e, d)$.

Based on the page view spike $S(e, d)$ of entity e for date d , the estimation of $P(d|e)$, which is further smoothed using Laplace smoothing, is given as

$$P(d|e) = \frac{S(e, d) + \kappa}{\sum_{d_i \in T} S(e, d_i) + \kappa|T|} \quad (5.9)$$

where $|T|$ is the number of days contained in the longest date range T supported by the system, which is set as one year here. Consequently, the probability $P(t|e)$ reflecting events about e happening within t can be calculated as follows (here we assume that the dates within t are independent given the entity e)

$$P(t|e) = \prod_{d_i \in t} P(d_i|e) \quad (5.10)$$

5.3.3. Relatedness Model

The distribution $P(e'|e, t)$ models the entity relatedness between e and e' w.r.t. t . To estimate $P(e'|e, t)$, we consider both static and dynamic entity relatedness as

$$P(e'|e, t) = \lambda \frac{R_S(e, e')}{\sum_{e'} R_S(e, e')} + (1 - \lambda) \frac{R_D(e, e', t)}{\sum_{e'} R_D(e, e', t)} \quad (5.11)$$

where $R_S(e, e')$ measures the *static relatedness* between e and e' , $R_D(e, e', t)$ measures the *dynamic relatedness* between e and e' w.r.t. t and λ is a parameter, which is set as 0.2 by default and will be discussed in detail in the experiments. For the special case that $e = e'$, we define $R_S(e, e') = R_D(e, e', t) = 1$.

For each pair of entities e and e' , we calculate their *static relatedness* $R_S(e, e')$ by adopting the Wikipedia link based measure introduced by (Milne & Witten, 2008a) as

$$R_S(e, e') = 1 - \frac{\log(\max(|E|, |E'|)) - \log(|E \cap E'|)}{\log(|W|) - \log(\min(|E|, |E'|))} \quad (5.12)$$

where E and E' are the sets of entities that link to e and e' respectively, and W is the set of all entities.

In order to measure the *dynamic relatedness* $R_D(e, e', t)$, we propose a novel approach based on *entity co-occurrence* in Web documents and *spike overlap* of page views, which will be discussed in the following.

Entity Co-occurrence. Based on the real-time stream of multilingual Web news articles annotated with entities, we investigate *entity co-occurrence* in the Web documents, which expresses the strength of dynamic entity association. For each pair of e and e' w.r.t. t , we calculate the entity co-occurrence measure $EC(e, e', t)$ by adopting the method of χ^2 hypothesis test introduced by (Bron et al., 2010) as

$$EC(e, e', t) = \frac{N(t)(C(e, e', t)C(\bar{e}, \bar{e}', t) - C(e, \bar{e}', t)C(\bar{e}, e', t))^2}{C(e, t)C(e', t)(N(t) - C(e, t))(N(t) - C(e', t))} \quad (5.13)$$

where $N(t)$ is the total number of Web documents published within the date range t , $C(e, e', t)$ denotes the co-occurrence frequency of e and e' in the Web documents within t , $C(e, t)$ and $C(e', t)$ denote the frequencies of e and e' occurring in the Web documents within t , respectively, and \bar{e}, \bar{e}' indicate that e, e' do not occur in Web documents, i.e., $C(\bar{e}, \bar{e}', t)$ is the number of documents within t where neither e nor e' occurs, and $C(e, \bar{e}', t)$ ($C(\bar{e}, e', t)$) denotes the number of documents within t where e (e') occurs but e' (e) does not.

Spike Overlap. Based on the page view spike of entities, we propose *spike overlap* $SO(e, e', t)$ to affect the dynamic relatedness between entities e and e' w.r.t. t . The intuition is that the page view spike of e and e' on the same date d will contribute to the dynamic relatedness between e and e' . In this regard, we calculate $SO(e, e', t)$ by adopting the weighted Jaccard similarity as

$$SO(e, e', t) = \frac{\sum_{d \in \mathcal{I}} \min\{S(e, d), S(e', d)\}}{\sum_{d \in t} \max\{S(e, d), S(e', d)\}} \quad (5.14)$$

where \mathcal{I} can be defined as the given date range t , i.e., $\mathcal{I} = t$. However, the above defined measure is only based on page view spikes of entities and thus suffers from the situation that entities with significant page view spike on the same date might not be associated in reality. Therefore, we construct the date set \mathcal{I} as

$$\mathcal{I} = \{d | C(e, e', d) \geq \tau, d \in t\} \quad (5.15)$$

where the co-occurrence frequency $C(e, e', d)$ of e and e' in the Web documents published on d has to exceed a threshold τ , which helps to determine if the page view spike overlap is more likely to indicate an association between e and e' than just by chance. Based on our observation, it is reasonable to set τ as 10.

By taking both *entity co-occurrence* in Web documents and *spike overlap* of page views into consideration, we calculate the *dynamic relatedness* $R_D(e, e', t)$ between entities e and e' for a specific date range t as follows

$$R_D(e, e', t) = EC(e, e', t) \cdot SO(e, e', t)^2 \quad (5.16)$$

5.3.4. Mention Model

The distribution $P(s|e)$ models the likelihood of observing the mention s given the intended entity e . To estimate $P(s|e)$, we employ Wikipedia and Wikilinks datasets and propose a point-wise mutual information (PMI) based method as

$$P(s|e) = \frac{PMI(e, s)}{\sum_{s_i \in S_e} PMI(e, s_i)} \quad (5.17)$$

where S_e is the set of surface forms of entity e and $PMI(e, s)$ is calculated as

$$PMI(s, e) = \log \frac{P(s, e)}{P(s)P(e)} = \log \frac{C(e, s) \times N}{C(s) \times C(e)} \quad (5.18)$$

where we have $P(s) = \frac{C(s)}{N}$, $P(e) = \frac{C(e)}{N}$, $P(s, e) = \frac{C(e, s)}{N}$ based on maximum likelihood estimation (MLE), $C(s)$ is the number of links using s as anchor text, $C(e)$ is the number of links pointing to e , $C(e, s)$ is the number of links using s as anchor text pointing to e and N is the total number of links.

5.3.5. Context Model

The probability $P(c|e, t)$ models the likelihood of observing the context c given the query entity e and the date range t . The context c of e contains the surface forms of other entities related to e . Assuming that all surface forms s_c in the context c are independent given e and t , the probability $P(c|e, t)$ is estimated as

$$P(c|e, t) = \prod_{s_c \in c} P(s_c|e, t) \quad (5.19)$$

The problem remains to estimate $P(s_c|e, t)$, the probability that a surface form s_c appears in the context of e w.r.t. t .

Given the query entity e and date range t , we consider a generation process of the context, where the context model first finds the *related entities* of e w.r.t. t based on the relatedness model, and then generates the surface form s_c of such related entities as the context of e based on the mention model. The form of the context generation for the query entity e and date range t is given as

$$P_R(s_c|e, t) = \sum_{e_{s_c} \in E_{s_c}} P(e_{s_c}, s_c|e, t) = \sum_{e_{s_c} \in E_{s_c}} \underbrace{P(e_{s_c}|e, t)}_{\text{Relatedness}} \underbrace{P(s_c|e_{s_c})}_{\text{Mention}} \quad (5.20)$$

where E_{s_c} denotes the set of entities having surface form s_c and we assume that s_c is independent of e and t given e_{s_c} , i.e., $P(s_c|e_{s_c}, e, t) = P(s_c|e_{s_c})$.

The above estimation suffers from the sparse data problem, i.e., some entities are not related to a given query entity e , but might appear as the context of e in the query q , which results in zero probability. Therefore, we perform smoothing by giving some probability mass to such unrelated entities. The general idea is that a surface form s_c of entities that are not related to the query entity e should also be possible to appear in the context of e and can be generated by chance. In this regard, we define the probability $P(s)$ of surface form s , which is built from the *entire collection* of entities and surface forms, as

$$P(s) = \frac{\sum_{e \in E_s} C(e, s)}{\sum_{s_i \in S} \sum_{e_i \in E_{s_i}} C(e_i, s_i)} \quad (5.21)$$

where S is the set of all surface forms, E_s is the set of entities having surface form s , and $C(e, s)$ denotes the frequency that s refers to e .

In order to achieve a robust estimation of the context model, we further smooth $P_R(s_c|e, t)$ using $P(s)$ based on Jelinek-Mercer smoothing as follows

$$P(s_c|e, t) = \gamma P_R(s_c|e, t) + (1 - \gamma)P(s_c) \quad (5.22)$$

where γ is a tunable parameter that is set to 0.9 by line search in our experiments. This estimation mixes the probability of s_c derived from the related entities of e with the general collection frequency of s_c used to refer to any entities.

5.4. Experiments

We now discuss the experiments we have conducted to assess the performance of our approach to TER based on our newly created benchmark datasets.

5. Time-Aware Entity Recommendation

Domain	Query	Time Range	Description of Information Needs
<i>Sports</i>	Germany Brazil	2014/07	The Germany and the Brazil football teams in the FIFA World Cup
<i>Entertainment</i>	Clooney	2014/09	The wedding of George Clooney
<i>Business</i>	Alibaba	2014/09	Alibaba Group making an IPO in the United States
<i>Emergencies</i>	Indonesia Java	2014/12	The Indonesia AirAsia Flight 8051 crashing into the Java Sea
<i>Society</i>	Pistorius	2014/09	Oscar Pistorius with nickname "Blade Runner" killing his girlfriend
<i>Science</i>	Rosetta	2014/11	The spacecraft Rosetta first successful landing on the comet
<i>Politics</i>	Donald Tusk	2014/12	Donald Tusk becoming the President of the European Council

Table 5.1.: Examples of information needs.

5.4.1. Experimental Setup

In our experiments, we employ DBpedia 2014⁶ as the knowledge base and the Wikipedia snapshot of June 2014 as the auxiliary data source. Existing datasets for the evaluation of entity recommendation aim to quantify the degree to which entities are related to the query without involving temporal aspects, which makes such datasets unsuitable for the TER task. There are some studies using a subset of TREC queries for time-aware information retrieval, where the goal is to investigate the user’s implicit temporal intent for document retrieval (Li & Croft, 2003; Kanhabua & Nørsvåg, 2010). However, such datasets do not contain the time ranges of interest explicitly given by users along with the queries and thus cannot be used for the TER evaluation. Therefore, we have created a new dataset where we asked 6 students, who also serve as judges of the experimental results, to provide information needs of both queries and date ranges. By removing the duplicate ones, it results in a final set of 22 information needs in different domains including Sports, Entertainment, Business, Emergencies, Society, Science and Politics. Some examples of such information needs are shown in Table 5.1. The datasets used in our experiments are available at <http://km.aifb.kit.edu/sites/ter/>.

To the best of our knowledge, no existing work on the TER task can be found. Therefore, we build the following baselines for comparison with our approach: (1) the first baseline is a static method using an ad hoc ranking function without considering the given time range t , defined as $Score(e, q) = \sum_{e_q} C(e_q) R_S(e_q, e)$, where $C(e_q)$ represents the commonness of each query entity e_q w.r.t. the corresponding mention in the query q , which has been introduced by (Milne & Witten, 2008b; Shen et al., 2012), and $R_S(e_q, e)$ denotes the Wikipedia link based relatedness between each query entity e_q and the candidate entity e (Milne & Witten, 2008a); (2) the second baseline is similar to our probabilistic model, but without taking into account the time range t , defined as $P(e, q) = \sum_{e_q} P(e)P(e_q|e)P(s_{e_q}|e_q)P(c_{e_q}|e_q)$, where $P(e)$ and $P(s_{e_q}|e_q)$ are estimated using our popularity and mention models respectively, $P(e_q|e)$ and $P(c_{e_q}|e_q)$ are also estimated using our relatedness and context models, but with $\lambda = 1$ (see Eq. 5.11), i.e., only the static relatedness between entities is considered in these models. For a comparative analysis, we have conducted the experiments with several

⁶<http://wiki.dbpedia.org/Downloads2014>

methods: the above described two baselines, denoted by *BSL1* and *BSL2*, respectively; our proposed method leaving out each of the popularity, temporality, relatedness, mention and context models, denoted by $-\Phi_e$, $-\Phi_{t|e}$, $-\Phi_{e'|e,t}$, $-\Phi_{s|e}$ and $-\Phi_{c|e,t}$, respectively; and our method with all these five models, denoted by *Full Model*.

The existing work, such as the Spark system from Yahoo! (Blanco et al., 2013) and the similar one published by Microsoft (Yu et al., 2014; Bi et al., 2015), could also be used for comparison with our method, even though they are not dedicated to the TER task. However, these systems assume that a query refers to only one entity, so they cannot deal with our more general case, where the query could involve multiple query entities. More importantly, these systems rely on the datasets that only major Web search engines have and are not publicly accessible. Due to these reasons, it is difficult to re-implement such systems and compare them with our method.

5.4.2. Results of Entity Retrieval

To assess the quality of entities retrieved by our method, we employ Normalized Discounted Cumulative Gain (nDCG) at rank k (Järvelin & Kekäläinen, 2000) as quality criteria, which is defined as $nDCG@k = \frac{DCG@k}{IDCG@k}$, where $DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i+1)}$ and rel_i is the graded relevance assigned to the result at position i and $IDCG@k$ is the maximum attainable $DCG@k$. This measure captures the goodness of a retrieval model based on the graded relevance of the top- k results. For each information need, all the entities retrieved by different methods are judged on 1-5 relevance scale by 6 students based on the criteria including both relevance and timeliness w.r.t. the underlying information needs. The final relevance of each candidate entity is determined by the relevance score voted by most judges and ties are resolved by the author. More details about the description of each graded relevance are available in our datasets.

The experimental results of $nDCG@k$ with varying k for different methods are shown in Table 5.2. Our method with *Full Model* performs the best for different k . Compared with the static baseline *BSL2* using a similar probabilistic model, it achieves 32.5%, 35.1%, 34.2%, 36.9% and 40.6% improvements when k is 5, 10, 15, 20 and 30, respectively. The baselines only obtain better results compared with our method without the relatedness model, while our method leaving out any other model still greatly outperforms the baselines. By comparing the two static baselines, *BSL2* clearly outperforms *BSL1*, which also shows the advantage of the method based on our probabilistic model over the ad hoc method.

As we focus on the TER task, the capability of our method to find temporally related entities is of great importance such that we have created an additional dataset consisting of only temporally related entities, which are also determined based on the votes of the 6 judges. Firstly, they are asked to select the entities that are temporally related to each information need and such entities are then ranked by the number of times being selected. Only the top-5 ranked

5. Time-Aware Entity Recommendation

	nDCG@k							<i>Full Model</i>
	<i>BSL1</i>	<i>BSL2</i>	$-\Phi_e$	$-\Phi_{t e}$	$-\Phi_{e' e,t}$	$-\Phi_{s e}$	$-\Phi_{c e,t}$	
$k=5$	0.597	0.622	0.805	0.778	0.140	0.800	0.797	0.824
$k=10$	0.594	0.621	0.817	0.786	0.176	0.803	0.804	0.839
$k=15$	0.596	0.640	0.846	0.810	0.505	0.830	0.823	0.859
$k=20$	0.616	0.642	0.865	0.831	0.521	0.853	0.847	0.879
$k=30$	0.635	0.658	0.898	0.877	0.552	0.895	0.887	0.925

Table 5.2.: $nDCG@k$ of retrieved entities (with the best results in bold).

	Recall@k							<i>Full Model</i>
	<i>BSL1</i>	<i>BSL2</i>	$-\Phi_e$	$-\Phi_{t e}$	$-\Phi_{e' e,t}$	$-\Phi_{s e}$	$-\Phi_{c e,t}$	
$k=5$	0.273	0.264	0.464	0.464	0.091	0.491	0.491	0.518
$k=10$	0.318	0.309	0.582	0.591	0.146	0.591	0.600	0.646
$k=15$	0.318	0.336	0.655	0.655	0.182	0.700	0.700	0.736
$k=20$	0.346	0.346	0.709	0.682	0.255	0.746	0.736	0.755
$k=30$	0.364	0.364	0.791	0.827	0.318	0.846	0.809	0.855

Table 5.3.: $Recall@k$ of temporally related entities (with the best results in bold).

candidates are included into the final dataset, where ties are resolved by the author. This results in 110 entities in total (5 for each of the 22 information needs).

In this experimental setting, we are concerned with whether these temporally related entities can appear on top of the ranked list of the retrieved entities. For this, we consider recall at rank k ($recall@k$) as quality criteria, where recall defines the number of relevant results that are retrieved in relation to the total number of relevant results and $recall@k$ is defined by only taking into account the top- k results. The experimental results of $recall@k$ with varying k for different methods are shown in Table 5.3. While the two static baselines exhibit only minor differences, our method with *Full Model* achieves a considerable performance improvement over the baselines for different k .

For both measures of $nDCG@k$ and $recall@k$, we observe that our method achieves better results by adding each individual model and the relatedness model that incorporates both static and dynamic entity relatedness contributes the most. For example, when $k = 30$, $nDCG@k$ and $recall@k$ decrease 40.1% and 62.8% respectively, by ablating the relatedness model, while the performance reduction without the other models ranges from 5.2% to 2.9% for $nDCG@k$ and from 7.5% to 1.1% for $recall@k$.

5.4.3. Results of Entity Ranking

The measures of $nDCG@k$ and $recall@k$ assess the quality of only top- k results, while we would like to evaluate the ranking of entities from highly relevant ones to only remotely rele-

<i>Gold Standard</i>	<i>BSL2</i>	<i>Full Model</i>
Germany national football team	Latin America	Brazil national football team
Brazil national football team	Brazil national football team	Germany national football team
2014 FIFA World Cup	Brazil national basketball team	2014 FIFA World Cup
Joachim Löw	2014 FIFA World Cup	Luiz Felipe Scolari
Toni Kroos	Germany national football team	FIFA World Rankings
Luiz Felipe Scolari	FIFA World Rankings	Toni Kroos
Neymar	Luiz Felipe Scolari	Neymar
FIFA World Rankings	Neymar	Joachim Löw
Latin America	Joachim Löw	Latin America
Brazil national basketball team	Toni Kroos	Brazil national basketball team

Table 5.4.: The gold-standard ranking of 10 entities (with dynamically related ones in bold) for the query “*Germany Brazil*” and the date range “*July 2014*” as well as the rankings by the baseline *BSL2* and our method with *Full Model*.

Domain (#Query)	<i>BSL1</i>	<i>BSL2</i>	$-\Phi_e$	$-\Phi_{t e}$	$-\Phi_{e' e,t}$	$-\Phi_{s e}$	$-\Phi_{c e,t}$	<i>Full Model</i>
<i>Sports (6)</i>	0.149	0.289	0.531	0.572	0.240	0.646	0.529	0.663
<i>Entertainment (4)</i>	0.191	0.252	0.594	0.645	0.188	0.667	0.673	0.688
<i>Business (3)</i>	0.596	0.596	0.790	0.834	-0.139	0.838	0.855	0.838
<i>Emergencies (4)</i>	-0.130	-0.082	0.473	0.421	0.470	0.503	0.467	0.494
<i>Others (5)</i>	0.365	0.358	0.612	0.522	0.232	0.576	0.527	0.581
Average	0.216	0.272	0.586	0.582	0.219	0.634	0.588	0.642

Table 5.5.: Spearman rank correlation between the gold-standard ranking and the ranking generated by different methods (with the best results in bold).

vant or even not relevant ones. Therefore, we have created another dataset, where the author select 10 candidate entities for each information need in a way that their relevances are clearly distinguishable among each other. Similar to (Hoffart et al., 2012), the gold-standard ranking of the 10 candidate entities is then created in the following way: (1) for all possible comparisons of the 10 candidate entities (45 in total), the 6 judges are asked which of the given two entities is more related to the information need by considering both relevance and timeliness; (2) all comparisons are then aggregated into a single confidence value for each entity and the 10 candidate entities are ranked by these confidence values as described by (Coppersmith et al., 2010). The final output is a set of 22 ranked lists consisting of 10 entities for each, against which we compare the automatically generated rankings by different methods using Spearman rank correlation, which measures the strength of association between two ranked variables. Some examples of different rankings are shown in Table 5.4.

The Spearman rank correlation between the gold-standard ranking and the automatically generated rankings by all these methods is given in Table 5.5. It shows that the experimental results of entity ranking are consistent with the results obtained in the entity retrieval experi-

5. Time-Aware Entity Recommendation

Domain	$\lambda = 0$	$\lambda = .1$	$\lambda = .2$	$\lambda = .3$	$\lambda = .4$	$\lambda = .5$	$\lambda = .6$	$\lambda = .7$	$\lambda = .8$	$\lambda = .9$	$\lambda = 1$
<i>Sports</i>	0.620	0.653	0.663	0.636	0.634	0.610	0.604	0.564	0.541	0.489	0.285
<i>Entertainment</i>	0.573	0.670	0.688	0.636	0.612	0.530	0.512	0.473	0.445	0.439	0.348
<i>Business</i>	0.737	0.838	0.838	0.842	0.842	0.842	0.822	0.826	0.834	0.794	0.657
<i>Emergencies</i>	0.530	0.518	0.494	0.509	0.467	0.479	0.458	0.412	0.367	0.303	-0.058
<i>Others</i>	0.537	0.576	0.581	0.564	0.537	0.503	0.505	0.534	0.537	0.493	0.280
Average	0.592	0.639	0.642	0.625	0.606	0.579	0.568	0.549	0.531	0.489	0.284

Table 5.6.: Spearman rank correlation between the gold-standard ranking and the ranking by our *Full Model* for different λ (with the best results in bold).

ments. The static baseline *BSL2* with a probabilistic model yields slightly better results than the baseline *BSL1* that is based on an ad hoc method. Clearly, our method with *Full Model* achieves the best results and considerably outperforms the baselines. Similarly, all the individual models contribute to the final performance improvement, where the relatedness model contributes the most. By respectively ablating the models Φ_e , $\Phi_{t|e}$, $\Phi_{e'|e,t}$, $\Phi_{s|e}$ and $\Phi_{c|e,t}$, the performance correspondingly reduces 8.7%, 9.3%, 65.8%, 1.2% and 8.4%.

Our method is sensitive to the parameter λ used in the relatedness model (see Eq. 5.11). Intuitively, a smaller λ reflects that the dynamic entity relatedness measure plays a more important role in the model. Table 5.6 shows the impact of λ on the ranking performance of our method with *Full Model*, where $\lambda = 0.2$ yields the best results on average, which has been used as the default value in our experiments. We observe that only using the dynamic relatedness measure, i.e., $\lambda = 0$, achieves the best results for the *Emergencies* domain. This is because in this domain there are more entities that are only dynamically related to the query. For example, given the information need about the crash of Indonesia AirAsia Flight 8501 into the Java sea in December 2014, where the query is “*Indonesia Java*” and the date range is “*December 2014*”, the related entities *AirAsia*, *Aviation_accidents_and_incidents* and *Search_and_rescue* do not have a static connection with the query. Another tunable parameter is γ (see Eq. 5.22). We observe that $\gamma = 0.9$ achieves the best results, which has been set as the default value in our experiments. For the sake of space, we omit the results based on different γ because they exhibit only minor differences.

5.5. Related Work

The TER task can be placed in the context of (1) entity search, (2) related entity recommendation and (3) temporal information retrieval.

Entity search has been defined by (Pound et al., 2010) as finding entities explicitly named in the query. Recently, entity search becomes more complex and closer to question answering when the query only provides a description of the target entity, where a list of member relationships to a single entity is given in the query. A recent development in evaluating entity

search of this type was the introduction of the Related Entity Finding using Linked Open Data (REF-LOD) task at the TREC Entity Track in 2010 and 2011 (Balog et al., 2011), where the type of relation to the target entity and the type of the target entity are both given as constraints.

For *related entity recommendation*, the Spark system developed at Yahoo! extracts several features from a variety of data sources and uses a machine learning model to produce a recommendation of entities to a Web search query, where neither the relation type nor the type of the target entity are specified (Blanco et al., 2013). Following Spark, Sundog aims to improve entity recommendation, in particular with respect to freshness, by exploiting Web search log data using a stream processing based implementation (Fischer et al., 2015). Microsoft has also developed a similar system that performs personalized entity recommendation by analyzing user click logs and entity pane logs (Yu et al., 2014; Bi et al., 2015).

In recent years, the time dimension has received a large share of attention in *temporal information retrieval* (Kanhubua et al., 2015). The temporal characteristics of queries (Dai & Davison, 2010) and dynamics of document content (Elsas & Dumais, 2010) have been leveraged in relevance ranking. The real-time information extracted from Twitter has been used to train learning to rank models (Dong et al., 2010). To improve Web search results, the temporal information has also been used for query understanding (Kulkarni et al., 2011) and auto-completion of queries (Shokouhi & Radinsky, 2012).

5.6. Conclusions

In this chapter, we introduce a novel task of *time-aware entity recommendation* (TER), since we argue that time-awareness should be a crucial factor in entity recommendation, which has not been addressed so far. To tackle this challenge, we propose a probabilistic model that aims to rank related entities according to a time-specific information need presented as a keyword query and a date range. The main contribution of our approach is that we decompose the TER task into several well defined probability distributions, each representing the context of a different component in the model. Through these components, heterogeneous entity knowledge extracted from different data sources that are publicly available on the Web can be incorporated into our model. Experimental results show that our method clearly outperforms approaches that are not context-aware, specifically when being time-agnostic.

5. *Time-Aware Entity Recommendation*

6. Query Rewriting for Keyword Search on Graphs

The problem of *rewriting keyword search queries* on graph data has been studied recently, where the main goal is to clean user queries by rewriting keywords as valid tokens appearing in the data and grouping them into meaningful segments. The main existing solution to this problem employs heuristics for ranking query rewrites and a dynamic programming algorithm for computing them. Based on a broader set of queries defined by an existing benchmark, we show that the use of these *heuristics does not yield good results*. In this chapter, we present a novel *probabilistic framework*, which enables the optimality of a query rewrite to be estimated in a more principled way. We show that our approach outperforms existing work in terms of effectiveness and efficiency w.r.t. query rewriting. More importantly, we provide the first results indicating query rewriting can indeed *improve overall keyword search* runtime performance and result quality.

6.1. Introduction

Keyword search on graph data has attracted large interest. It has proven to be an intuitive and effective paradigm for accessing information, helping to circumvent the complexity of structured query languages and to hide the underlying data representation. Using simple keyword queries, users can search for complex structured results, including connected tuples from relational databases, XML data, RDF graphs, and general data graphs (Kacholia et al., 2005; He et al., 2007; Tran et al., 2009). Existing work so far focuses on the efficient *processing of keyword queries* (Hristidis et al., 2003; He et al., 2007), or effective *ranking of results* (Liu et al., 2006; Luo et al., 2007).

In addition, recent work studies the problem of *keyword query cleaning* (Pu & Yu, 2008; Gao et al., 2011). The motivation is *keyword queries are dirty*, often containing words not intended to be part of the query, words that are misspelled, or words that do not directly appear but are semantically equivalent to words in the data. Besides dirty queries, keyword search solutions also face the problem of *search space explosion*. Searching results on graph data requires finding matches for the individual keywords as well as considering subgraphs in the data connecting them, which represent final answers covering all query keywords. The space of possible subgraphs is generally exponential in the number of query keywords. Through

grouping keywords into larger meaningful units (called *segments*), the number of keywords to be processed and the corresponding search space is reduced.

The two main tasks involved in query cleaning (henceforth also called *query rewriting*) are *token rewriting*, where query keywords are rewritten as tokens appearing in the data, and *query segmentation*, where tokens are grouped together as segments representing compound keywords. Query rewriting helps to improve not only the result quality but also the runtime performance of keyword search. Towards a rewriting solution that enables more effective and efficient keyword search, we provide the following contributions:

Probabilistic Ranking of Query Rewrites and Its Impact on Keyword Search Effectiveness. The optimality of query rewrites has been defined based on heuristics for scoring tokens and segments, including an adoption of TFIDF (Pu & Yu, 2008; Gao et al., 2011). However, we show in this work that for ranking query rewrites, existing work based on these heuristics has several conceptual flaws and does not yield high quality results. Instead of using ad-hoc heuristics, we propose a probabilistic framework for keyword query rewriting, which enables the optimality of query rewrites to be studied in a systematic fashion. In particular, optimality is captured in terms of the probability a query rewrite can be observed given the data, and estimated using a principled technique (Maximum Likelihood Estimation). Furthermore, while previous work only considers the textual information but neglects the rather rich graph structure, which might be more crucial for keyword search on graph data, our approach takes both textual and structural information in the data into account. In (Pu & Yu, 2008; Gao et al., 2011), it has been shown that w.r.t. the proposed ad-hoc notion of optimality, computed rewrites are accurate. However, the actual effect of query rewriting on the quality of keyword search results is not clear. Using the recently established benchmark (Coffman & Weaver, 2010) for keyword search, we show that our approach not only yields *better query rewrites* but more importantly, also *better keyword search results*.

Context-based Computation of Query Rewrites and Its Impact on Keyword Search Efficiency. The problem of computing query rewrites has shown to be NP-hard. A solution (Pu & Yu, 2008) based on dynamic programming has been proposed for this, which computes optimal query rewrites by considering all possible combinations of optimal sub-query rewrites. There, the optimality of a rewrite is based on the optimality of all its components, while our probabilistic approach enables optimality to be captured merely based on the previously observed context in an incremental rewriting process. We show that this probabilistic model not only produces higher quality results but also can be exploited by a context-based top- k algorithm that is *more efficient* than the previous solution. Moreover, while previous work reported the search space reduction resulting from segmentation, its impact on overall keyword search performance is not clear. In this work, we show that the search space reduction can outweigh the overhead incurred through query rewriting, resulting in *better overall runtime performance*.

The rest of the chapter is organized as follows. We provide an overview of the problems

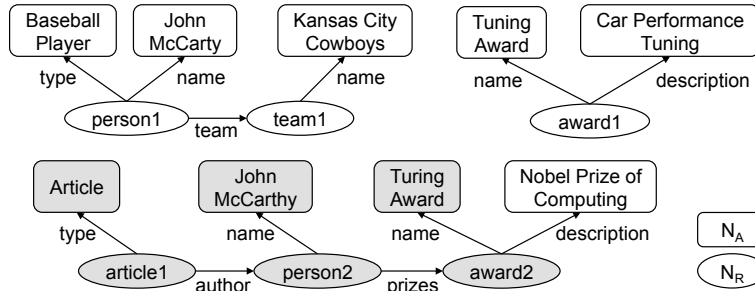


Figure 6.1.: Example data graph fragments from different sources covering three domains, i.e., baseball, cars and computer science.

in Sec. 6.2. Then, we present our solution for ranking and computing query rewrites along with differences to the most related work in Sec. 6.3 and Sec. 6.4, respectively. Experimental results are presented in Sec. 6.5, followed by more related work in Sec. 6.6 and conclusions in Sec. 6.7.

6.2. Overview

We firstly provide an overview of the keyword search problem, then discuss the role of keyword query rewriting.

6.2.1. Keyword Search on Graph Data

Keyword search solutions have been proposed for dealing with different kinds of data, including relational, XML and RDF data. In the general setting, existing approaches treat these different kinds of data as graphs:

Definition 5 (Data). *Data are captured as a directed labeled graph $D(N, E)$ called data graph, where $N = N_R \uplus N_A$ is the disjoint union of resource and attribute value nodes N_R and N_A , respectively, and $E = E_R \uplus E_A$ is the set of directed edges, where E_R are edges between two resources called relations, i.e., $e(n_i, n_j) \in E_R$ iff $n_i, n_j \in N_R$, and E_A are edges between a resource and an attribute value called attributes, i.e., $e(n_i, n_j) \in E_A$ iff $n_i \in N_R \wedge n_j \in N_A$. Each data element $e \in N \uplus E$ is labeled with some text $L(e)$ called label describing e .*

Results in this setting are defined as Steiner trees (Kacholia et al., 2005), or *Steiner graphs* in the graph data setting (Li et al., 2008; Ladwig & Tran, 2011):

6. Query Rewriting for Keyword Search on Graphs

Keyword Query	Possible Query Rewrites
“ <i>Publication John McCarthy Turing Award</i> ”	Article⊗John⊕McCarthy⊗Turing⊕Award* Article⊗John⊕McCarthy⊗Tuning⊕Award Article⊗John⊕McCarthy⊗Turing⊕Award Article⊗John⊕McCarthy⊗Tuning⊕Award

Table 6.1.: Possible query rewrites.

Definition 6 (Result / Steiner Graph). A result to a keyword query Q also called Steiner graph is a subgraph of $D(N, E)$ denoted as $D_S = (N_S, E_S)$, which satisfies the following conditions: 1) for every $q \in Q$ there is at least one element $n_q \in N$ (called keyword element) that matches q , i.e., the label $L(n_q)$ contains q . The set of keyword elements containing one for every $q \in Q$ is $N_Q \subseteq N_S$; 2) for every possible pair $n_i, n_j \in N_Q$ and $n_i \neq n_j$, there is a path $n_i \rightsquigarrow n_j$, i.e., an edge $e(n_i, n_j) \in E$ or a sequence of edges $e(n_i, n_k) \dots e(n_l, n_j)$ in E , such that every $n_i \in N_Q$ is connected to every other $n_j \in N_Q$. Such a graph is called a d -length Steiner graph when paths that connect keyword elements are of length d or less.

Example 1. Given the data graph in Fig. 6.1, for the keyword query shown in Table 6.1, there is one matching Steiner graph as highlighted in Fig. 6.1, namely the one connecting the three nodes *Article*, *John McCarthy* and *Turing Award* (assuming that keywords have already been rewritten so that they match the labels of these three nodes, e.g., “*Tuning Award*” has been rewritten to match the node *Turing Award*).

For finding whether some data elements match query keywords, existing solutions typically use an inverted index and treat elements (their labels) as documents (task 1). For finding paths to form Steiner graph from these elements (task 2), they explore the data as an undirected graph, traversing the edges without taking their direction into account. For pragmatic reasons, existing keyword search solutions (He et al., 2007; Tran et al., 2009; Li et al., 2008) apply a maximum path length restriction d , such that only paths of length d or less have to be traversed.

6.2.2. Keyword Query Rewriting

The label $L(e)$ of each data element e and the query Q can be conceived as a sequence of tokens, e.g., the label *Turing Award* consists of two tokens *Turing* and *Award*. Query rewriting firstly maps query keywords (also called query tokens) to tokens appearing in the labels of data elements (*token rewriting*), and then groups the resulting data tokens into segments to form *query rewrites* (*query segmentation*):

Definition 7 (Token Rewrite). Let TOKEN^D be the set of all tokens in the data graph D . Token rewriting with factor m is a function rewrite_m , which maps a query token q to a list of m data tokens $t \in \text{TOKEN}^D$ associated with the respective distance d between q and t . Given

a keyword query $Q = \{q_1, q_2, \dots, q_n\}$, a query token rewrite is a $m \times n$ matrix M of tokens $t \in \text{TOKEN}^D$, where the i -th column is obtained through $\text{rewrite}_m(q_i)$.

Example 2. Given the data graph in Fig. 6.1, we can construct the matrix M for the example query in Table 6.1 using the rewriting function rewrite_2 :

$$M = \begin{pmatrix} \text{Article} & \text{John} & \text{McCarty} & \text{Tuning} & \text{Award} \\ -- & -- & \text{McCarthy} & \text{Turing} & -- \end{pmatrix}$$

Note that the matrix M might have empty entries when there are less than m candidate data tokens for a query token.

Definition 8 (Segment and Query Rewrite). Given the query token rewrite M of dimension $m \times n$, a segment is a sequence of tokens in M from adjacent columns. A query rewrite (also called segmentation) is a sequence of continuous and non-overlapping segments $S = s_1 s_2 \dots s_k$ such that for all segments s_i , $1 \leq i \leq k$, the first column of s_{i+1} is next to the last column of s_i , i.e., $\text{start}(s_{i+1}) = \text{end}(s_i) + 1$, where $\text{start}(s)$ and $\text{end}(s)$ denote the first column and the last column covered by s , respectively. A query rewrite can also be seen as a sequence of tokens $t \in M$ and actions α , namely $S = t_1 \alpha_1 t_2 \dots t_{n-1} \alpha_{n-1} t_n$, where t_i denotes one token in the i -th column of M and α_i represents a concatenation action denoted by \oplus or a splitting action denoted by \circledast . A splitting action \circledast captures the boundary of two segments.

Example 3. For our example query, Table 6.1 shows a few query rewrites. The segment-based representation of the rewrite $\text{Article} \circledast \text{John} \oplus \text{McCarthy} \circledast \text{Tuning} \oplus \text{Award}$ is $s_1 = \{\text{Article}\}$, $s_2 = \{\text{John}, \text{McCarthy}\}$, $s_3 = \{\text{Tuning}, \text{Award}\}$.

Note that the first rewrite in the table captures the query we would like to obtain because it yields the Steiner graph presented in the previous example. As opposed to the original query, segments in this rewrite correspond to tokens in the data, thus facilitating the finding of relevant results. Further, because segments stand for compound query keywords, this rewrite contains only three instead of five. Observe that we have three other rewrites, where all constituent segments also correspond to data tokens. However, we can see data elements matching these segments are not connected, i.e., do not form Steiner graphs. We consider a rewrite to be *valid* when it yields Steiner graphs, and *relevant*, when these graphs represent relevant answers. In order to assess the relevance of answers, we use manually defined ground truth provided by the keyword search benchmark (Coffman & Weaver, 2010). Considering query rewrite optimality under these aspects of validity and relevance makes our work different from the main existing solution (Pu & Yu, 2008), which defines optimality based on several heuristics that we will discuss next.

6.3. Probabilistic Query Rewriting

Existing work (Pu & Yu, 2008) ranks a query rewrite S based on the sum of all the scores of its segments, where the score of each segment s depends on several heuristics, including the *distance* of tokens in s from the corresponding query keywords and the number of tokens in s . A central heuristic is the one based on an adoption of TFIDF. The TFIDF score of a segment s is defined as $Score_{IR}(s) = \max\{tfidf(s, e) : e \in N \uplus E\}$, where $tfidf(s, e)$ is the TFIDF weight of the segment s in the data element e , which is a tuple in previous work. With respect to the two main aspects of query rewriting, namely validity and relevance, we identify the following problems with TFIDF-based ranking:

Relevance. Intuitively, the TFIDF weight of a query term q is high for a document d , when d contains a large number of mentions of q (TF), and q discriminates d well from other documents (IDF). The adoption of TFIDF here computes the weight w.r.t. a tuple. However, query rewrites have to be ranked, not tuples. A query rewrite S may contain several segments corresponding to several tuples. Thus, when S contains a segment s with high TFIDF weight w.r.t. some tuples, it does not mean that S contains a large number of mentions of s and that s discriminate S well from others. In other words, it is not clear why a rewrite S with higher TFIDF weighted segments is more relevant.

Validity. The TFIDF heuristic and others do not consider structural information in the data. Some data elements contain tokens and segments that represent relevant candidates for token rewriting and segmentation. However, these elements only help to generate valid query rewrites, when they are actually parts of some Steiner graphs. Thus, to ensure validity, paths in the data have to be considered.

6.3.1. Probabilistic Model

Let $Q = \{q_1, q_2, \dots, q_n\}$ be the user query, D be the data, and $S = t_1 \alpha_1 t_2 \dots \alpha_{n-1} t_n$ be a query rewrite. The probability $P(S|Q, D)$ can be calculated based on Bayes theorem:

$$P(S|Q, D) = \frac{P(Q|S, D) \cdot P(S|D)}{P(Q|D)} \quad (6.1)$$

Since $P(Q|D)$ can be considered as a constant, denoted as γ , given the fixed Q and D , we have

$$P(S|Q, D) = \frac{1}{\gamma} \cdot P(Q|S, D) \cdot P(S|D) \quad (6.2)$$

The term $P(S|D)$ is of particular interest in this work, as it captures the *probability of query rewrites*. For token rewriting, we can focus on $P(Q|S, D)$, which captures the probability of observing (the keywords in) Q given the (tokens in the) intended query rewrite S and the data D .

6.3.2. Probabilistic Token Rewriting

Since users having the intended token t_i in mind specify the query keyword q_i commonly according to their word usage and spelling habit, we assume that each q_i is only related to the corresponding token rewrite t_i reflecting the user's search intention and the keyword query Q is independent of the data D given the intended query rewrite S , i.e., $P(Q|S, D) = P(Q|S)$. For the purpose of token rewriting, the actions in a query rewrite S can be removed and each q_i is only dependent on t_i . That is,

$$\begin{aligned} P(Q|S) &= P(q_1, q_2, \dots, q_n | t_1 \alpha_1 t_2 \dots \alpha_{n-1} t_n) \\ &= P(q_1, q_2, \dots, q_n | t_1, t_2, \dots, t_n) = \prod_{i=1}^n P(q_i | t_i) \end{aligned} \quad (6.3)$$

where $P(q_i | t_i)$ models the likelihood of observing a query keyword q_i , given that the intended token is t_i .

Then, this probability mass is distributed inverse proportionally to the distance $d(q_i, t_i)$, which measures the syntactic and semantic distance between q_i and t_i . In our implementation, $d(q_i, t_i)$ is a combination of edit distance and semantic distance, which is derived from the lexical database WordNet. For each query keyword q_i , we have

$$P(q_i | t_i) = \frac{1}{\varepsilon} \cdot \exp(-\eta \cdot d(q_i, t_i)) \quad (6.4)$$

where η is a parameter that controls how fast the probability decreases with the distance and ε is a normalization factor.

6.3.3. Probabilities of Query Rewrites

For query segmentation, S is conceived as a sequence of segments, or a *sequence of token and segmentation action pairs*, such that the probability $P(S|D)$ is estimated based on tokens and actions in S :

$$\begin{aligned} P(S|D) &= P(t_1 \alpha_1 t_2 \dots \alpha_{n-1} t_n | D) \\ &= \prod_{i=0}^{n-1} P_D(\alpha_i t_{i+1} | t_1 \alpha_1 t_2 \dots \alpha_{i-1} t_i) \end{aligned} \quad (6.5)$$

where $P_D(\alpha_0 t_1) = P_D(t_1)$ and $P_D(\alpha_i t_{i+1} | t_1 \alpha_1 t_2 \dots \alpha_{i-1} t_i)$ stands for $P(\alpha_i t_{i+1} | t_1 \alpha_1 t_2 \dots \alpha_{i-1} t_i, D)$. However, for a keyword query Q containing many keywords, computing $P(S|D)$ will incur prohibitive cost when D is large in size. To address this problem, we make the N^{th} order Markov assumption to approximate that the probability of

6. Query Rewriting for Keyword Search on Graphs

an action on a token only depends on the N preceding token and action pairs (to be precise, N preceding tokens and $N - 1$ actions and $N = 2$ in the following examples). That is,

$$P(S|D) \approx \prod_{i=0}^{n-1} P_D(\alpha_i t_{i+1} | t_{i-N+1} \alpha_{i-N+1} \dots \alpha_{i-1} t_i) \quad (6.6)$$

For computing this, we build upon the idea behind the *n-gram language model*. The *n-gram* model defines the probability of a sequence of tokens $s = t_1 t_2 \dots t_l$ that appear in the data as the joint probability of observing every token t_{i+1} in s , given the previous tokens $t_{i-N+1} \dots t_i$ (called context), i.e., $P(t_1 t_2 \dots t_l) \approx \prod_{i=0}^{l-1} P(t_{i+1} | t_{i-N+1} \dots t_i)$ (note that instead of n , we use N where $n = N + 1$). For various information retrieval and text processing tasks, this approximation based on the Markov assumption has proven to work well. We also rely on this assumption to *focus only on the previously observed context* during the computation of query rewrite probability. Typically, the Maximum Likelihood Estimation is employed, which computes this probability as the count of $t_{i-N+1} \dots t_i t_{i+1}$, divided by the sum of counts of all n -grams that share the same context $t_{i-N+1} \dots t_i$, i.e., $P(t_{i+1} | t_{i-N+1} \dots t_i) = \frac{C(t_{i-N+1} \dots t_i t_{i+1})}{\sum_t C(t_{i-N+1} \dots t_i t)}$, where $C(t_i \dots t_j)$ denotes the count of $t_i \dots t_j$ appearing in the data.

For query segmentation, we need to adopt this idea such that instead of token probability, the *action-token pair* probability specified in Eq. 6.6 can be derived. First, since query segmentation is order insensitive, i.e., both ‘‘John McCarthy’’ and ‘‘McCarthy John’’ should be grouped into one segment, we consider *n-gram* as a set of tokens that co-occur in a window of size n instead of a sequence of n tokens that appear contiguously. To facilitate the following discussion, we firstly define the concept of action induced segment:

Definition 9 (Action Induced Segment). *For $Q = \{q_1, q_2, \dots, q_n\}$ and the corresponding query rewrite $S = t_1 \alpha_1 t_2 \dots \alpha_{n-1} t_n$, a segment s_i induced by action α_{i-1} is the concatenation of the previously induced segment s_{i-1} resulting from α_{i-2} and the token t_i , i.e., $s_i = s_{i-1} t_i$ if $\alpha_{i-1} = \oplus$; otherwise (i.e., $\alpha_{i-1} = \circ$), $s_i = t_i$. For α_0 , we have $s_1 = t_1$. The induced segment $s_i(l)$ is a segment with length (i.e., the number of constituent tokens) no larger than l . For a segment s_i with more than l tokens, $s_i(l)$ is s_i without the first $l(s_i) - l$ tokens, where $l(s_i)$ is the length of s_i .*

While the *n-gram* model predicts the probability of a token t_{i+1} given the context s_i , the task of query segmentation is to predict the action-token pair $\alpha_i t_{i+1}$, i.e., the probability that t_{i+1} is concatenated with s_i ($\oplus t_{i+1}$) or that t_{i+1} forms a new segment ($\circ t_{i+1}$). Whereas \oplus depends on the probability t_{i+1} can be observed given s_i , the action \circ intuitively depends on the probability t_{i+1} has a different context $\neg s_i (\neq s_i)$. To compute the probabilities for both these actions, the entire event space consisting of both contexts s_i and $\neg s_i$ has to be taken into

account. Based on these observations, for the case where $i > 0$, we have

$$P_D(\alpha_i t_{i+1} | s_i(N), \neg s_i(N)) = \begin{cases} \frac{C(s_i(N)t_{i+1})}{\sum_t C(s_i(N)t) + C(\neg s_i(N)t)} & \text{if } \alpha_i = \oplus \\ \frac{C(\neg s_i(N)t_{i+1})}{\sum_t C(s_i(N)t) + C(\neg s_i(N)t)} & \text{if } \alpha_i = \circ \end{cases} \quad (6.7)$$

where $C(s_i(N)t_{i+1})$ is the count of $s_i(N)t_{i+1}$ as n -gram in the labels of some elements in D . Note that $\sum_t C(s_i(N)t) + C(\neg s_i(N)t) = \sum_t C(t)$. For $i = 0$, the query rewrite probability can be computed by considering only the first token because there is no need to make an action. Thus, we have

$$P_D(\alpha_0 t_1) = P_D(t_1) = \frac{C(t_1)}{\sum_t C(t)} \quad (6.8)$$

where $C(t)$ is the count of token t in D . The following example shows that while intuitively appealing, using this probability of query rewrite leads to unexpected results.

Example 4. Suppose that for the partial keyword query $Q' = \text{“Publication John McCarthy”}$ we have $S' = \text{Article} \circ \text{John} \oplus \text{McCarthy}$. Given the next query keyword “Tuning” , we then have the token rewrites “Tuning” and “Turing” , and the counts $C((\text{John} \oplus \text{McCarthy}) \text{Tuning}) = 0$ and $C((\text{John} \oplus \text{McCarthy}) \text{Turing}) = 0$ because “Tuning” and “Turing” never appear together with “John McCarthy” , $C(\neg(\text{John} \oplus \text{McCarthy}) \text{Tuning}) = 2$ and $C(\neg(\text{John} \oplus \text{McCarthy}) \text{Turing}) = 1$ because “Tuning” and “Turing” appear respectively twice and once in other contexts. Based on Eq. 6.7, we have $P(\circ \text{Tuning} | \text{John} \oplus \text{McCarthy}) = \frac{2}{3}$ and $P(\circ \text{Turing} | \text{John} \oplus \text{McCarthy}) = \frac{1}{3}$. The resulting query rewrites are respectively $\text{Article} \circ \text{John} \oplus \text{McCarthy} \circ \text{Tuning}$ and $\text{Article} \circ \text{John} \oplus \text{McCarthy} \circ \text{Turing}$, where the former is more likely than the latter. Continuing with “Award” , we obtain 4 final query rewrites where those with “Tuning” still have higher probability than those with “Turing” . Looking at the data, we rather expect the contrary, i.e., those with “Turing” should be preferred.

6.3.4. Probabilities of Valid Query Rewrites

The previous model considers relevance but not validity. The probability of every action-token pair $\alpha_i t_{i+1}$ depends on the count of $\neg s_i(N)t_{i+1}$. This may lead to cases, where query rewrites do not yield Steiner graphs, i.e., the segments match keyword elements that are not connected. In particular, the previous example show that $P(\circ \text{Tuning} | \text{John} \oplus \text{McCarthy})$ is relatively high (i.e., relevant) because Tuning matches some data elements. However, $\text{John} \oplus \text{McCarthy}$ and Tuning match data elements that are not connected and thus the splitting action inducing $\text{John} \oplus \text{McCarthy} \circ \text{Tuning}$ does not result in any answer (i.e., is not valid).

6. Query Rewriting for Keyword Search on Graphs

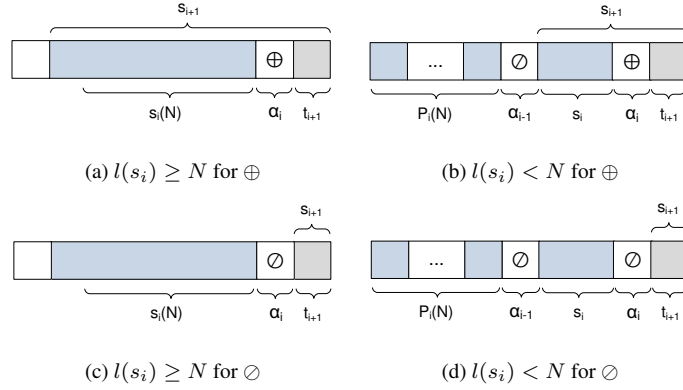


Figure 6.2.: Segment s_{i+1} induced by action α_i performed on segment s_i (set of segments $\mathcal{P}_i(N)$) and token t_{i+1} .

The above problem arises because the language model is designed to model unstructured data. It might be ineffective when applied to Steiner graphs, which are rich in structural information. Extending this model to take the graph structure into account, we propose to focus on estimating the actions only based on events that actually lead to results. The goal is to produce *valid query rewrites*, which yield non-empty sets of Steiner graphs. Clearly, it follows from Def. 6 that a query rewrite is valid when every possible pair of its segments is connected. More formally, the connectivity of segments is defined as follows:

Definition 10 (Connected Segments). *Let s_i and s_j be two segments, $N_i, N_j \subseteq N$ be the sets of corresponding keyword elements in $D(N, E)$ such that for each $n_i \in N_i$ and $n_j \in N_j$, the labels $L(n_i)$ and $L(n_j)$ contain s_i and s_j , respectively. The segments s_i and s_j are connected (denoted as $s_i \leftrightarrow s_j$) when there is at least one $n_i \in N_i$ and one $n_j \in N_j$ and $n_i \neq n_j$ such that $n_i \leftrightarrow n_j$, where the d -length restriction of paths also applies.*

With the N^{th} order Markov assumption, there are two cases to consider for computing valid query rewrites. When the previously induced segment s_i has length equal or greater than N , it suffices to focus on $s_i(N)$ to predict the next action α_i on t_{i+1} . Fig. 6.2(a) and 6.2(c) illustrate this, showing the induced segment s_{i+1} given the action α_i is \oplus or \emptyset . As before, the events for $\oplus t_{i+1}$ are $s_i(N)t_{i+1}$ (clearly, these events lead to valid segments because they correspond to cases where elements in the data graph have labels containing $s_i(N)t_{i+1}$). In cases where t_{i+1} does not have context $s_i(N)$, \emptyset is considered. However, \emptyset only yields Steiner graphs when t_{i+1} is connected with $s_i(N)$. That is, instead of all $\neg s_i(N)t_{i+1}$, only the events $s_i(N) \leftrightarrow t_{i+1}$ are relevant in this case. Note that $\neg s_i(N)t_{i+1}$ captures all events where t_{i+1} does not co-occur with $s_i(N)$, which clearly include all events where t_{i+1} appears in the label $L(n_i)$, $s_i(N)$ appears in the label $L(n_j)$ and $n_i \neq n_j$. The set of events denoted by $s_i(N) \leftrightarrow t_{i+1}$ is a subset of events captured by $\neg s_i(N)t_{i+1}$, namely $n_i \leftrightarrow n_j$ instead of

$n_i \neq n_j$. We use $s_i(N) \leftrightarrow t_{i+1}$ to focus on valid query rewrites while $\neg s_i(N)t_{i+1}$ stands for all query rewrites. For estimating the probability, we have

$$P_D(\alpha_i t_{i+1} | s_i(N), s_i(N) \leftrightarrow) = \begin{cases} \frac{C(s_i(N)t_{i+1})}{\sum_t C(s_i(N)t) + C(s_i(N) \leftrightarrow t)} & \text{if } \alpha_i = \oplus \\ \frac{C(s_i(N) \leftrightarrow t_{i+1})}{\sum_t C(s_i(N)t) + C(s_i(N) \leftrightarrow t)} & \text{if } \alpha_i = \circ \end{cases} \quad (6.9)$$

As opposed to the previous adoption of the n -gram model, focusing on s_i alone when it has length less than N is not enough. This is because the connectivity of segments induced previous to s_i has an impact on the validity of query rewrites. The action α_i on the next token t_{i+1} depends on the set of previously induced segments $\mathcal{P}_i(N)$ and s_i , where $\mathcal{P}_i(N)$ is the set of the induced segments that precede s_i and together with s_i , contains at most N tokens in total, i.e., $\sum_{s_{\rho_i} \in \mathcal{P}_i(N)} l(s_{\rho_i}) + l(s_i) \leq N$. The components to be considered for the probability estimation of \oplus and \circ are shown in Fig. 6.2(b) and 6.2(d), respectively. The segment $s_i t_{i+1}$ resulting from the concatenation action \oplus is valid only when $s_i t_{i+1}$ is connected to all preceding segments in $\mathcal{P}_i(N)$. Similarly, a splitting action \circ only leads to valid segments when t_{i+1} is connected to all preceding segments in $\mathcal{P}_i(N) \cup \{s_i\}$ (henceforth, simply denoted as $\mathcal{P}_i(N)s_i$). Thus in this case, the probability is estimated as

$$P_D(\alpha_i t_{i+1} | \mathcal{P}_i(N) \leftrightarrow s_i, \mathcal{P}_i(N)s_i \leftrightarrow) = \begin{cases} \frac{C(\mathcal{P}_i(N) \leftrightarrow s_i t_{i+1})}{\sum_t C(\mathcal{P}_i(N) \leftrightarrow s_i t) + C(\mathcal{P}_i(N)s_i \leftrightarrow t)} & \text{if } \alpha_i = \oplus \\ \frac{C(\mathcal{P}_i(N)s_i \leftrightarrow t_{i+1})}{\sum_t C(\mathcal{P}_i(N) \leftrightarrow s_i t) + C(\mathcal{P}_i(N)s_i \leftrightarrow t)} & \text{if } \alpha_i = \circ \end{cases} \quad (6.10)$$

where $C(\mathcal{P} \leftrightarrow s)$ denotes the count of segment s that is connected to all segments in the set of segments \mathcal{P} .

In addition to these two cases, Eq. 6.8 also applies for the case $i = 0$, because no actions have to be considered.

Example 5. Consider the same case as in Example 4, for Q' , S' and the next query keyword “Tuning”. Due to the same reason, we have $C((John \oplus McCarthy) Tuning) = 0$ and $C((John \oplus McCarthy) Turing) = 0$. Differently, we observe that $C((John \oplus McCarthy) \leftrightarrow Tuning) = 0$ and $C((John \oplus McCarthy) \leftrightarrow Turing) = 1$, because “Tuning” is connected with “John McCarthy” once but “Tuning” never. Based on Eq. 6.9, we have $P(\circ Tuning | John \oplus McCarthy) = 1$. Accordingly, the only query rewrite with non-zero probability is $Article \circ John \oplus McCarthy \circ Turing$. When continuing with the keyword “Award”, instead of a total of 4 final query rewrites, only the valid query rewrite $Article \circ John \oplus McCarthy \circ Turing \oplus Award$ remains.

6.3.5. Reward Maximization Framework

Besides this principled ranking model based on language modeling, additional heuristics that may perform well in specific settings can be added on top using a reward model. A typical assumption in keyword search is that when a result is more compact, it is considered to be more meaningful and relevant (Kacholia et al., 2005). Also, neighboring query keywords should be grouped together to produce longer segments (Pu & Yu, 2008).

We propose a reward model to accommodate heuristics. A reward is associated with every action made in the query rewriting process. To give preference to longer segments for instance, we assign a reward for each action α_i as

$$R(\alpha_i) = \exp(\beta \cdot l(s_{i+1})) \quad (6.11)$$

where s_{i+1} is the segment induced by α_i , β is used to control the importance of this length based heuristic and $R(\alpha_0) = 1$.

The *overall reward* of a query rewrite S is computed from the rewards of all actions made during query rewriting, i.e.,

$$R(S) = \prod_{i=0}^{n-1} R(\alpha_i) \quad (6.12)$$

The final ranking, which combines the probability of query rewrites $P(S|Q, D)$ with the additional quality criteria $R(S)$, is captured by the *conditional reward* defined as

$$\begin{aligned} R(S|Q, D) &= R(S) \cdot P(S|Q, D) = \frac{1}{\gamma} \cdot R(S) \cdot P(Q|S) \cdot P(S|D) \\ &= \frac{1}{\gamma} \cdot \prod_{i=0}^{n-1} R(\alpha_i) \cdot P(q_{i+1}|t_{i+1}) \cdot P_D(\alpha_i t_{i+1}|e) \end{aligned} \quad (6.13)$$

where $e = null$ when $i = 0$, $e = \{s_i(N), s_i(N) \leftrightarrow\}$ when $l(s_i) \geq N$, and $e = \{\mathcal{P}_i(N) \leftrightarrow s_i, \mathcal{P}_i(N) s_i \leftrightarrow\}$ when $l(s_i) < N$. Now, we arrive at our final notion of optimality:

Definition 11. (Optimal Query Rewrites). *Given the data D , the query Q and its set of query rewrites \mathcal{S} , the optimal query rewrite S^* is the one with the highest conditional reward, i.e., $S^* = \arg \max_{S \in \mathcal{S}} R(S|Q, D)$. The top- k optimal query rewrites \mathcal{S}^k are the k ones with the highest conditional rewards.*

6.4. Computing Top-k Query Rewrites

We will briefly revisit existing work on query rewriting and show that our model enables a more efficient algorithm by focusing only on the previously observed *context*. First, we present the indexes and then the top- k rewriting algorithm.

Segment	Segment Length	Count
s_i	$l(s_i) \leq N + 1$	$C(s_i)$
$s_i \rightsquigarrow S_j$	$l(s_j) + \sum_{s \in S_j} l(s) \leq N + 1$	$C(s_i \rightsquigarrow S_j)$

Table 6.2.: The extended n -gram index capturing segments containing no more than $N + 1$ tokens, and their connections.

6.4.1. Indexing

For token rewriting, tokens are managed separately in a *token index*. It keeps tokens in the data as well as semantically related entries such as synonyms extracted from WordNet. The semantic distance between them is precomputed and stored. This and the edit distance between query and index tokens are used to compute $P(q_{i+1}|t_{i+1})$ in Eq. 6.13.

For query segmentation, we build an *extended n -gram index* to materialize segments and connections between them. It stores all segments s_i containing no more than $N + 1$ tokens and their counts. Further, let s_j denote segments that have length less than $N + 1$. For every s_i , the set of all possible combinations of segments connected to s_i that together with s_i have total length no more than $N + 1$, denoted as S_j , are stored in the index together with the count of $s_i \rightsquigarrow S_j$. The extended n -gram index is illustrated in Table 6.2. This index is employed to compute $P_D(\alpha_i t_{i+1}|e)$ in Eq. 6.13.

For efficient extended n -gram indexing, we employ the concept of *connectivity matrix* M_D^d , which is a boolean matrix capturing paths between nodes in the data graph D . An entry m_{ij}^d in M_D^d is 1, iff there is a path between nodes n_i and n_j of length no larger than d ; otherwise, m_{ij}^d is 0. The matrix M_D^d is constructed iteratively using the formula $M_D^d = M_D^{d-1} \times M_D^1$. These matrices can be represented by tables of the maximum size n_c containing connected node pairs in D , where n_c denotes the number of node pairs that are connected by paths of length d or less. M_D^d is then generated by performing join on M_D^{d-1} and M_D^1 . For further details, we refer the interested readers to (Tran & Zhang, 2014).

Now we clarify the index costs of our approach. Let n_a , n_r and $n = n_a + n_r$ be the number of attribute value nodes, resource nodes and all nodes in D respectively, and l be the bound of their labels. The time complexity and index size w.r.t. the token index are both $\mathcal{O}(n_a \cdot l)$. For constructing the extended n -gram index, nodes in the data graph have to be joined for computing paths between them. In the worst case, a join on $input_i$ and $input_j$ requires $|input_i| \times |input_j|$ time such that the complexity of computing paths with length no larger than d is $\mathcal{O}(n_c^2 \cdot d)$. In practice, join operation can be performed more efficiently using special indexes and implementations like hash join. As a result, instead of $|input_i| \times |input_j|$, a join requires only $|input_i| + |input_j|$ such that the complexity is $\mathcal{O}(n_c \cdot d)$. Clearly, there are at most $\mathcal{O}(n_a \cdot l)$ segments s_i resulting in the time complexity and index size both as $\mathcal{O}(n_a \cdot l)$. For each s_i , at most $\mathcal{O}((n_{a \rightsquigarrow}^{max} \cdot l)^N)$ combinations of connected segments S_j can be found, where $n_{a \rightsquigarrow}^{max}$ denotes the maximum number of attribute value nodes that are connected with

one and the same attribute value node by paths. As this has to be done for all segments, the complexity for processing them is $\mathcal{O}(n_a \cdot l \cdot (n_{a \leftrightarrow}^{max} \cdot l)^N)$. Accordingly, the index size w.r.t. the connected segments also comes to $\mathcal{O}(n_a \cdot l \cdot (n_{a \leftrightarrow}^{max} \cdot l)^N)$.

In summary, the total time complexity w.r.t. the construction of the extended n -gram index is $\mathcal{O}(n_c \cdot d + n_a \cdot l + n_a \cdot l \cdot (n_{a \leftrightarrow}^{max} \cdot l)^N)$, including time for join processing and time for indexing the individual segments s_i and the connected segments $s_i \leftrightarrow S_j$. The total index size is $\mathcal{O}(n_a \cdot l + n_a \cdot l \cdot (n_{a \leftrightarrow}^{max} \cdot l)^N)$, including both indexes of s_i and $s_i \leftrightarrow S_j$. In our experiments, we use $N = 2$ ($n = 3$), which has shown to be sufficient for queries used in the benchmark (Coffman & Weaver, 2010). Additionally, while $n_{a \leftrightarrow}^{max} = n_a$ and $n_c = n^2$ at the most, in practice they are likely to be relatively small, as one node is not connected to all others but only a limited number of them, especially given the maximum path length d , such that the overall time complexity and index size are much smaller than the worst case. Compared with the indexing of previous work (Pu & Yu, 2008), which has the time complexity and index size both as $\mathcal{O}(n_a \cdot l)$, our indexing process is still more expensive. However, the additional indexing consumption will become the supplementary to the online query processing, which we will discuss later.

6.4.2. Holistic Top-k Query Rewriting

Previous work (Pu & Yu, 2008) has shown that the problem of computing top-k query rewrites is NP-hard and proposed a dynamic programming solution, which relies on a procedure for computing the top- k segments ($\text{find_}s^k$). The input is the token rewrite matrix M of dimension $m \times n$ (n denotes number of query keywords and m the number of tokens for every keyword). For any given (sub-)query covering keywords from i to j , $\text{find_}s^k$ computes the optimal segments $s^k(i, j)$ that cover the columns from i to j in M . A greedy algorithm is employed for scanning paths in the submatrix of dimension $m \times n'$, $n' = j - i + 1$, which in the worst case, produces $m^{n'}$ possible segments. The complexity of $\text{find_}s^k$ is $\mathcal{O}(m^l)$, when assuming that the lengths of database terms, namely labels, are bounded by l and $l < n'$, otherwise $\mathcal{O}(m^{n'})$.

Clearly, query rewriting solution ($\text{find_}S^k$) covering the columns from i to j may include optimal segments of length n' as well as any combination of smaller segments in sub-solutions that spans from i to j for finding the top- k rewrites $S^k(i, j)$, which results in the complexity of $\mathcal{O}(k \cdot n' \cdot m^l)$. For computing rewrites of a query of length n , we need to find the optimal segments of length n , as well as solving (a maximum of n^2) sub-problems of finding and combining query rewriting solutions of (sub-)queries covering keywords from i to j , $1 \leq i \leq j \leq n$, such that the complexity of computing top- k query rewrites is $\mathcal{O}(k \cdot n^3 \cdot m^l)$.

Fig. 6.3(a) illustrates the *bottom-up approach*, where each box with label *Token Rewrite* denotes a set of tokens in the data for each keyword and each box with label *Segment* and *Query Rewrite* stands for a set of optimal segments and rewriting solutions for a particular pair of

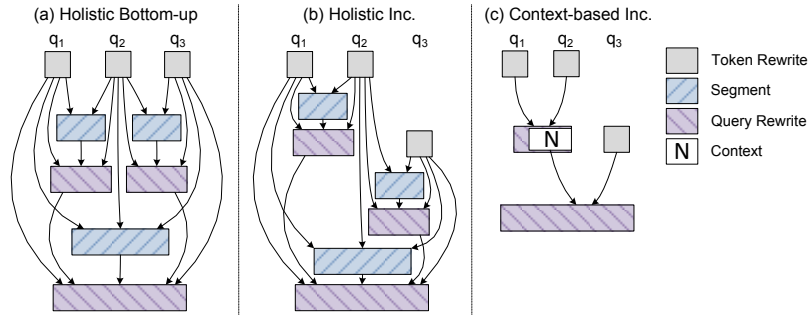


Figure 6.3.: Approaches to query rewriting.

(i, j) , respectively. For computing the *Query Rewrite* box corresponding to $i = 1$ and $j = 3$, which represents the solution to the final query consisting of three keywords, this approach starts with smaller solutions and iteratively combines them (the combination is illustrated through arrows). The *incremental variant* of this approach is shown in Fig. 6.3(b), which involves solving the same (number of) subproblems. The difference is only the order in which the sub-solutions are combined (it incrementally covers more keywords in every iteration). An early return condition is introduced, which can yield $\mathcal{O}(k \cdot n^2 \cdot m^l)$ but because there is no guarantee for this to apply, the worst case complexity is still $\mathcal{O}(k \cdot n^3 \cdot m^l)$.

6.4.3. Context-based Top-k Query Rewriting

A substantial difference between previous work and ours lies in the notion of optimal query rewrites. The previous algorithm takes all possible segments of a (sub-)query rewrite into account because determining optimality requires computing the score of every (sub-)query rewrite, which is based on the score of all its segments. As opposed to that, our probabilistic model provides a principled way to compute query rewrite scores based on query rewrites probabilities, and to *focus only on the previously observed context*.

We propose an incremental top- k procedure that starts with query rewrites containing one token and then iteratively constructs larger query rewrites by appending more token rewrites. Fig. 6.3(c) illustrates that query rewrites in each iteration are computed based on the combination of query rewrites obtained in the previous iteration and token rewrites from the current iteration. The main difference to the holistic approach is that in each iteration, instead of considering all combinations of sub-solutions as well as the segments covering the current query, we directly employ the previous query rewrites. In particular, we focus on those ones that vary in the context of a fixed length N (because intuitively speaking, only this context has an impact on the optimality). We introduce the notion of *pattern* to group query rewrites representing the same context.

6. Query Rewriting for Keyword Search on Graphs

Definition 12 (Prefix, Suffix and Pattern). *Given a (partial) query rewrite $S = t_1\alpha_1t_2 \dots \alpha_{n-1}t_n$, a prefix of S with length l is a partial query rewrite $\hat{S} = t_1\alpha_1t_2 \dots \alpha_{l-1}t_l$ and a suffix of S with length l is a partial query rewrite $\bar{S} = t_{n-l+1}\alpha_{n-l+2}t_{n-l+2} \dots \alpha_{n-1}t_n$, where $1 \leq l \leq n$. The pattern p of a query rewrite S is the suffix of S with length N , when S has more than N tokens, otherwise p is S .*

When partial query rewrites share the same pattern, the one with higher conditional reward is preferred over one other because it results in final rewrites with higher rewards:

Lemma 1. *Let $Q = Q'Q''$ consisting of two partial queries Q' and Q'' . Let S'' be a query rewrite corresponding to Q'' , S'_1, S'_2 and $S_1 = S'_1\alpha S''$, $S_2 = S'_2\alpha S''$ be two particular query rewrites corresponding to Q' and Q , respectively. When S'_1 and S'_2 share the same pattern, i.e., $p(S'_1) = p(S'_2)$, we have*

$$R(S'_1|Q', D) > R(S'_2|Q', D) \implies R(S_1|Q, D) > R(S_2|Q, D)$$

Proof Outline: Consider $l(Q'') = 1$. For any rewrite of Q'' denoted by t (i.e., $S'' = t$), we have conditional rewards $R(S_1|Q, D) = \frac{1}{\gamma} \cdot R(S'_1|Q', D) \cdot [R(\alpha) \cdot P(Q''|t) \cdot P_D(\alpha t|e_1)]$ and $R(S_2|Q, D) = \frac{1}{\gamma} \cdot R(S'_2|Q', D) \cdot [R(\alpha) \cdot P(Q''|t) \cdot P_D(\alpha t|e_2)]$ using Eq. 6.13. When S'_1 and S'_2 have the same pattern p , the events are same, i.e., $e_1 = e_2$ such that $P_D(\alpha t|e_1) = P_D(\alpha t|e_2)$. Hence, if $R(S'_1|Q', D) > R(S'_2|Q', D)$, then we have $R(S_1|Q, D) > R(S_2|Q, D)$. This also generalizes to $l(Q'') > 1$. For $Q'' = \{q_j, \dots, q_n\}$, we have $R(S_1|Q, D) = \frac{1}{\gamma} \cdot R(S'_1|Q', D) \cdot \prod_{i=j-1}^{n-1} [R(\alpha_i) \cdot P(q_{i+1}|t_{i+1}) \cdot P_D(\alpha_i t_{i+1}|e_{i,1})]$ and $R(S_2|Q, D) = \frac{1}{\gamma} \cdot R(S'_2|Q', D) \cdot \prod_{i=j-1}^{n-1} [R(\alpha_i) \cdot P(q_{i+1}|t_{i+1}) \cdot P_D(\alpha_i t_{i+1}|e_{i,2})]$. Because $e_{i,1} = e_{i,2}$ for $j-1 \leq i < n$.

We not only prefer the ones with higher rewards but, more specifically, we can focus on the k ones with highest rewards. We provide this theorem to capture that it is sufficient to keep track of the top- k rewrites for each distinct pattern:

Theorem 1. *Let $Q = (q_1, \dots, q_n)$ be the query and $Q' = (q_1, \dots, q_i)$ be any partial query s.t. $0 < i < n$. Let \mathbf{S}^k be the top- k query rewrites of Q and \mathbf{S}_p^{ik} be those top- k query rewrites of Q' with pattern p . Then for any non-top- k query rewrite $S'_p \notin \mathbf{S}_p^{ik}$ with pattern p , there is no top- k query rewrite $S \in \mathbf{S}^k$ such that S'_p is a prefix of S .*

Proof Outline: Assume that there is a top- k query rewrite $S = S'_p\alpha S''$ of Q with a non-top- k S'_p as prefix. Let $\bar{S} = \bar{S}'_p\alpha S''$ be a query rewrite of Q with a top- k $\bar{S}'_p \in \mathbf{S}_p^{ik}$ as prefix. As $R(\bar{S}'_p|Q', D) > R(S'_p|Q', D)$, it follows from Lemma 1 that $R(\bar{S}|Q, D) > R(S|Q, D)$. Thus, there are at least k query rewrites \bar{S} with $R(\bar{S}|Q, D) > R(S|Q, D)$, which contradicts the assumption that S is a top- k rewrite.

Algorithm 1: Finding Top-k Query Rewrites

Input: the user query $Q = \{q_1, q_2, \dots, q_n\}$.
Result: the top- k optimal query rewrites \mathbf{S}^k .

```

1  $P \leftarrow \emptyset$ ;
2 foreach  $i \in [1 \dots n]$  do
3    $(P', P) \leftarrow (P, \emptyset)$ ;
4    $T_i \leftarrow \text{TokensRewrites}(q_i)$ ;
5   foreach  $sp \in \text{CommonSubpatterns}(P')$  do
6      $P'_{sp} \leftarrow \text{PatternsWithSuffix}(P', sp)$ ;
7     foreach  $t \in T_i$  do
8        $\mathbf{S}^k_{sp \oplus t} \leftarrow \cup_{p' \in P'_{sp}} (\mathbf{S}^k_{p'} \bowtie_{\oplus} t)$ ;
9       if  $\mathbf{S}^k_{sp \oplus t} \neq \emptyset$  then
10        |  $P \leftarrow P \cup \{sp \oplus t\}$ ;
11        end
12         $\mathbf{S}^k_{sp \circ t} \leftarrow \cup_{p' \in P'_{sp}} (\mathbf{S}^k_{p'} \bowtie_{\circ} t)$ ;
13        if  $\mathbf{S}^k_{sp \circ t} \neq \emptyset$  then
14         |  $P \leftarrow P \cup \{sp \circ t\}$ ;
15         end
16        end
17      end
18    end
19   $\mathbf{S}^k \leftarrow \cup_{p \in P} \mathbf{S}^k_p$ ;
20 return  $\mathbf{S}^k$ ;

```

6.4.4. Algorithm

Based on these results, we propose an algorithm that in every iteration, joins token rewrites with previous partial query rewrites and keeps the top- k results for each pattern.

Definition 13 (Action Induced Join). *Let \mathbf{S} be a set of partial query rewrites with non-zero rewards, i.e., $\forall S \in \mathbf{S}, R(S|Q, D) > 0$. The join between \mathbf{S} and a token t induced by an action α results in a new set of query rewrites*

$$\mathbf{S} \bowtie_{\alpha} t = \{S\alpha | S \in \mathbf{S} \wedge R(S\alpha|Q, D) > 0\}$$

Performing this join thus requires computing the reward for $S\alpha t$ (via Eq. 6.13). The \bowtie_{α} is only successful when adding t to S (through concatenation or splitting) does not render the resulting rewrite invalid, i.e., only when $R(S\alpha t|Q, D) > 0$.

Definition 14 (Top-k Union). *Given sets of rewrites $[\mathbf{S}] = \{\mathbf{S}_1, \dots, \mathbf{S}_m\}$, where every rewrite in \mathbf{S}_i is associated with a reward, the top- k union $\cup_{\mathbf{S}_i \in [\mathbf{S}]}^k \mathbf{S}_i$ simply returns the k rewrites in $\cup_{\mathbf{S}_i \in [\mathbf{S}]} \mathbf{S}_i$ with the highest rewards.*

Employing these operators, Alg. 1 starts with the first query keyword ($i = 1$) and iteratively constructs larger rewrites by appending more keywords ($1 < i \leq n$). It uses P' and P to keep track of the patterns of the last and current iteration, and S_p^k to keep track of the top- k rewrites for each pattern p . In every iteration, $sp\alpha t$ are collected (line 10 and 14) and added to P , where sp is a subpattern and t a token rewrite. A subpattern sp of p is simply p when $l(p) < N$, otherwise it is $p(N - 1)$ (p without the first token). The grouping of patterns in P' to their subpatterns sp (line 6) yields group containing elements $p' \in P'_{sp}$ that share the same suffix sp . For each q_i , a list T_i of m token rewrites are retrieved from the token index (line 4). For every subpattern sp and $t \in T_i$, the new patterns $sp \oplus t$ and $sp \otimes t$ can be formed. For each new pattern, the top- k query rewrites $S_{sp\alpha t}^k$ are computed and updated by employing \bowtie_α and top- k union (line 8 and 12). The final top- k query rewrites of Q are computed by applying the top- k union on the top- k results S_p^k obtained for each $p \in P$ (line 19).

Complexity. In each iteration, there are at most m token rewrites, which have to be joined with the k results for each pattern. In the worst case, the number of patterns is same as the number of segments of length N , which as discussed, is m^N . As this has to be done for n iterations, the total complexity of Alg. 1 is $\mathcal{O}(k \cdot n \cdot m^{N+1})$. With respect to the complexity of the holistic approach, $\mathcal{O}(k \cdot n^3 \cdot m^l)$, using previously obtained query rewrites in every iteration and focusing on the context of length N translate to the changes from n^3 to n and m^l to m^{N+1} . The former can yield a substantial difference in performance because while the other parameters can be fixed to a small number, the number of keywords n cannot be controlled and may be large. The latter effect can also be substantial as it has been shown that n -grams with a relatively small N are indeed sufficient in many information and text processing tasks, while the bound of labels l could be much larger.

6.5. Experiments

We performed experiments to assess the merits of our approach to query rewriting and its impact on keyword search based on the recently established benchmark (Coffman & Weaver, 2010).

6.5.1. Experimental Setup

We compare our approach with an implementation of the state-of-the-art keyword query cleaning solution (*BQR*) (Pu & Yu, 2008). We use two variants of our approach, one ranks based on the probability of query rewrites (*PQR*) and the other uses the probability of valid query rewrites (*PVQR*) as discussed in Sec. 6.3.3 and Sec. 6.3.4. Both of them integrate the additional heuristics shown in Sec. 6.3.5. All systems were implemented in Java 1.6 on top of

Dataset	Size	Rel.	Tuples	I_{Token}	I_{PQR}	I_{PVQR}	I_{BQR}
Mondial	9	28	17,115	0.2/0.04	0.3/0.08	1.8/0.18	2.2/0.05
IMDb	516	6	1,673,074	7.9/8.67	179/17.6	303/40.8	150/9.53
Wikipedia	550	6	206,318	13/2.18	320/3.46	445/8.01	176/2.26

Table 6.3.: Dataset size, number of relations and tuples, index size/indexing time w.r.t. token index I_{Token} (same one used by all approaches) and the additional indexes used by two variants of our approach I_{PQR} , I_{PVQR} and the one used by the state-of-the-art baseline I_{BQR} (all sizes and time are in MB and minutes).

Dataset	$ Q $	$ q $	$ \bar{q} $	$ R $	$ \bar{R} $
Mondial	50	1-5	2.04	1-35	5.90
IMDb	50	1-26	3.88	1-35	4.32
Wikipedia	50	1-6	2.66	1-13	3.26

Table 6.4.: Number of queries $|Q|$, range in number of query keywords $|q|$ and relevant results $|R|$, average number of query keywords $|\bar{q}|$ and relevant results $|\bar{R}|$ per query.

MySQL¹ and Lucene². Experiments were performed on a Linux server with two Intel Xeon 2.8GHz Dual-Core CPUs and 8GB memory. We use all the three sets of data, queries, and relevance assessments available in the benchmark (Coffman & Weaver, 2010). In the experiments, we use $N = 2$ and $d = 3$, which are sufficient for queries used in the benchmark. We found that the setting of $\eta = 1$, $\beta = 0.33$ and $m = 10$ achieve the best performance. All reported results are based on these values. The effects of these model parameters are discussed in detail in Sec. 6.5.3.

Data. Table 6.3 provides the main statistics of the three datasets. IMDb employed in (Coffman & Weaver, 2010) is actually a subset from the original IMDb. Also, a selection of articles from Wikipedia was included in the benchmark, and the PageLinks table was augmented with an additional foreign key to explicitly indicate referenced pages. The Mondial dataset is much smaller, which captures geographical and demographic information from the Web sources such as the CIA World Factbook.

Indexes. Table 6.3 also reports indexing performance of the three systems w.r.t. index size and indexing time. As shown, the index used by PVQR needs more time and space than the one for PQR, because the former indexes not only n -grams, but also connectivity information. Compared to BQR, PVQR’s index is about a factor of 2 larger and the indexing process takes about 4 times longer, which is consistent with our analysis. We also provide a breakdown of the indexing time of PVQR into two parts attributable to join processing and

¹<http://www.mysql.com>

²<http://lucene.apache.org>

6. Query Rewriting for Keyword Search on Graphs

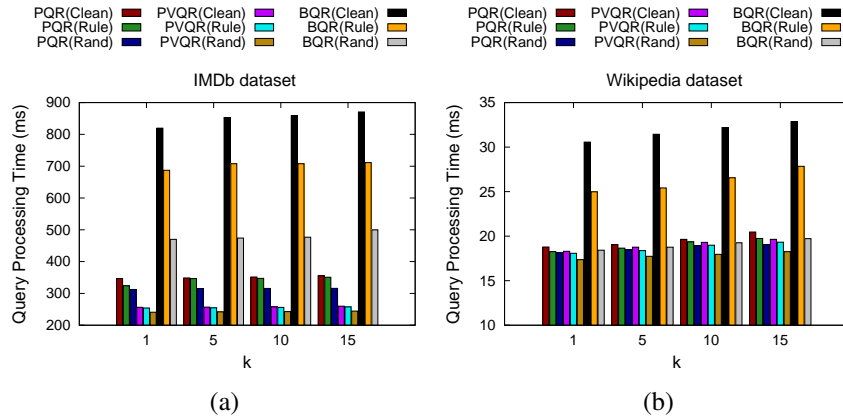


Figure 6.4.: Evaluation results for efficiency of query rewriting.

index creation. The elapsed time (join processing + index creation) for indexing Mondial, IMDb and Wikipedia is respectively 0.18(0.06+0.12), 40.8(11.5+29.3) and 8.01(0.47+7.54) minutes. Observe that join processing make up 33%, 28% and 6% of the indexing time for Mondial, IMDb and Wikipedia, respectively. The reason of such difference lies in the graph topology of the datasets, where the structure of Mondial is slightly more complex than that of IMDb, which in turn is much denser than that of Wikipedia.

Queries. For each dataset, 50 queries were proposed (Coffman & Weaver, 2010). Table 6.4 provides the statistics of queries and results. Many keywords in these queries can be grouped into segments. While they are suitable for studying the segmentation problem, further token modifications are needed to study token rewriting. From these queries, called *Clean* set, we obtain queries with dirty tokens by rewriting keywords following the same method used in XClean (Lu et al., 2011), a recent proposal for the token rewriting problem. First, we apply random edit operations, namely insertion, deletion and substitution, to each keyword with length larger than 4 in the Clean queries to obtain the *Rand* set of dirty queries. Second, we make use of the list of common misspellings occurring in Wikipedia³. For each Clean query, we replace the keyword that can be found in the list with one of its misspelled forms to obtain the *Rule* set of dirty queries.

6.5.2. Efficiency of Query Rewriting

Figs. 6.4(a-b) show the average time for computing top- k query rewrites for IMDb and Wikipedia. Mondial is a very small dataset, where all queries can be rewritten in less than

³http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

8 ms on average. For the sake of space, we omit its results because individual times exhibit only minor differences.

Compared to Wikipedia, IMDb contains many more tuples and IMDB queries are longer. This is reflected in the performance results. All systems take substantially more time for IMDb than Wikipedia. The performance of *PVQR* is consistently better than the other two systems for both datasets. *PVQR* is about 3-4 times faster than BQR for IMDb and about 2 times faster for Wikipedia. These differences are primarily due to the pruning capability of *PVQR*, i.e., *PVQR* prunes non-valid results. Compared to PQR, the amount of valid sub-query rewrites that have to be kept track of is smaller. The amount of partial rewrites considered by BQR is even much larger than PQR, as it considers all possible combinations of previously obtained segments. It is worth mentioning that Fig. 6.4(a) excludes the effect of 4 long IMDB queries with length 9, 11, 26, and 11. The reason is that BQR could not finish them within the time limit we set to 1 minute, while *PVQR* only takes 634 ms, 691 ms, 1657 ms and 746 ms respectively, for Clean queries (and even less for Rule and Rand queries).

We observe that *Clean queries require more time* than Rule queries, which in turn, take more time than Rand queries. This may seem less intuitive for that one would expect processing clean queries should be easier. Clearly, for Clean queries, the list of token rewrites always contains the intended one. These correct token rewrites yield segments, i.e., intermediate results, which have to be processed. For dirty queries, especially Rand, the list of token rewrites may contain no (or not many) correct ones, which cannot be combined to form segments, hence there are no (or fewer) intermediate results to be processed. *More time is needed for segmentation* when there are more intermediate results.

6.5.3. Effectiveness of Query Rewriting

The ground truth for this experiment can be obtained from the keyword search results captured by the mentioned benchmark. According to the results judged as correct, we add segment boundaries to the Clean queries. These *target queries* and their identified segments constitute the ground truth. This ground truth thus reflects both the quality of token rewriting and query segmentation. We use the standard metric *Mean Reciprocal Rank (MRR)* and an adaption of *Precision at k (P@k)*. Given a set of keyword queries \mathcal{Q} and the corresponding top- k lists of rewrites, let \mathcal{Q}^* be the queries for which the correct rewrite could be captured by the corresponding top- k list, and for each query $Q_i \in \mathcal{Q}$, let $rank_i$ be the rank of the correct rewrite in the top- k list, then $P@k = \frac{|\mathcal{Q}^*|}{|\mathcal{Q}|}$ and $MRR = \frac{1}{|\mathcal{Q}|} \sum_{Q_i \in \mathcal{Q}} \frac{1}{rank_i}$.

First, we study the effects of different model parameters on query rewriting. We experimented with different values of η , which reflects the sensitivity to spelling errors and semantic differences (see Eq. 6.4). The effect of η on MRR values for *PVQR* is shown in Table 6.5. The best results are highlighted in bold font. Observe that $\eta = 1$ achieves the best results for almost every query set except Clean queries for IMDb. The MRR values increase quickly from $\eta = 0$

6. Query Rewriting for Keyword Search on Graphs

Query Set	$\eta = 0$	1	5	10	15
Mondial(Clean)	0.80	0.97	0.97	0.97	0.97
Mondial(Rule)	0.80	0.97	0.97	0.97	0.97
Mondial(Rand)	0.86	0.99	0.99	0.99	0.97
IMDb(Clean)	0.67	0.82	0.83	0.83	0.83
IMDb(Rule)	0.68	0.82	0.81	0.81	0.81
IMDb(Rand)	0.60	0.77	0.73	0.72	0.72
Wikipedia(Clean)	0.81	0.94	0.94	0.94	0.94
Wikipedia(Rule)	0.79	0.89	0.89	0.89	0.89
Wikipedia(Rand)	0.84	0.93	0.91	0.91	0.91

Table 6.5.: MRR of PVQR vs. η ($\beta = 0.33$).

Query Set	$\beta = 0$	0.25	0.33	0.5	1
Mondial(Clean)	0.97	0.97	0.97	0.97	0.97
Mondial(Rule)	0.97	0.97	0.97	0.97	0.97
Mondial(Rand)	0.98	0.99	0.99	0.99	0.99
IMDb(Clean)	0.74	0.80	0.82	0.82	0.81
IMDb(Rule)	0.74	0.80	0.82	0.82	0.81
IMDb(Rand)	0.68	0.74	0.77	0.77	0.77
Wikipedia(Clean)	0.89	0.94	0.94	0.94	0.93
Wikipedia(Rule)	0.85	0.89	0.89	0.89	0.89
Wikipedia(Rand)	0.88	0.92	0.93	0.93	0.93

Table 6.6.: MRR of PVQR vs. β ($\eta = 1$).

to $\eta = 1$, then reach a plateau. When $\eta > 1$, while the MRRs might increase slightly for clean queries (see IMDb(Clean)), we observe minor decrease for dirty queries (see IMDb(Rule), IMDb(Rand) and Wikipedia(Rand)). This is probably due to the fact that when η is higher, we are stricter with the distance between token rewrites and query keywords. In other words, we prefer the original queries without token rewriting. That has a beneficial effect on clean queries but might bring errors in dirty queries because the misspelled query keywords will be ranked higher. The effect of β , which reflects the sensitivity to the length of segment (see Eq. 6.11), for PVQR is shown in Table 6.6. When β is larger, longer segments are preferred. In the experiments, the MRRs improve when β is larger than 0. This means applying this segment length based heuristic yields better results. However, this should not be done too aggressively: the best results are achieved when β reaches 0.33. To study the effect of m , which denotes the number of token rewrites considered for each query keyword, we vary its value from 1 to 15. We observe that the MRR values for all three approaches are highest and most stable when m approaches 10.

Fig. 6.5(a) illustrates MRR for the three datasets. Similar to the performance results, IMDb

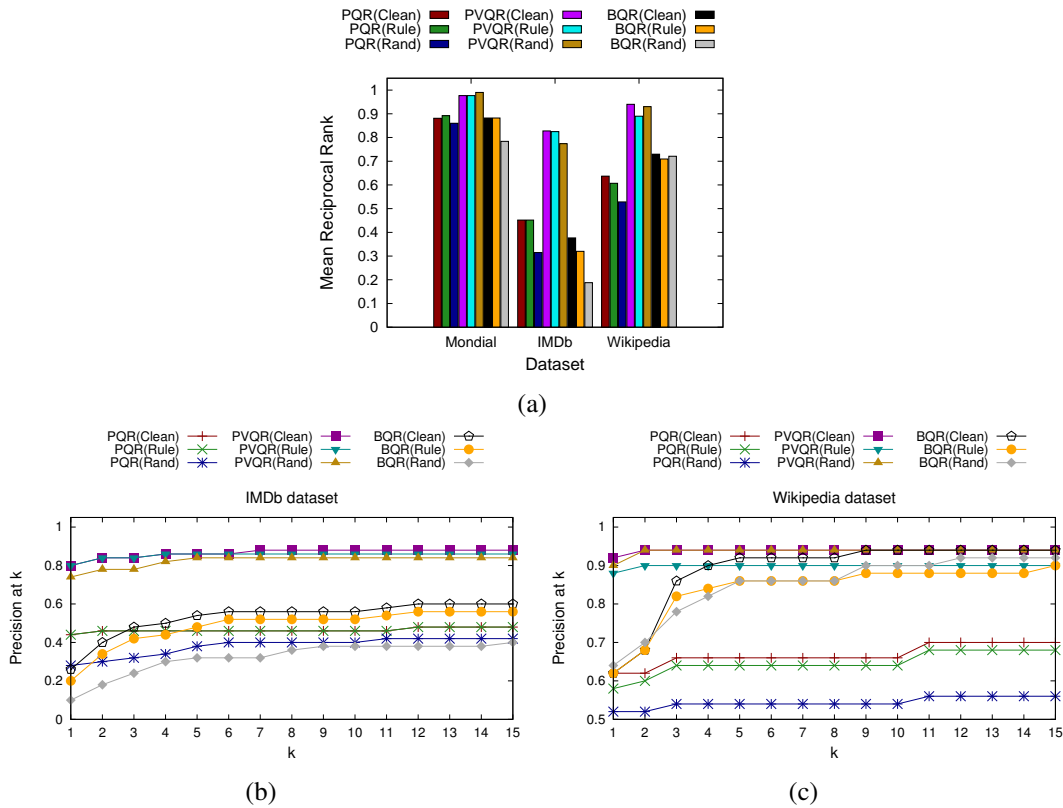


Figure 6.5.: Evaluation results for effectiveness of query rewriting.

constitutes the most difficult case, where MRR is particularly low for PQR and BQR. PVQR achieves the best results for all types of queries over all datasets. On average, *Rand queries* yield the lowest MRR while *Clean queries* the highest. This is expected because in the latter case, it is easier to obtain correct token rewrites, hence more relevant segments can be constructed.

Figs. 6.5(b-c) illustrate $P@k$ for IMDb and Wikipedia. On average, PVQR also outperforms the other two systems for all types of queries. For Wikipedia, BQR achieves good results when k is large, especially for Clean queries. Nevertheless, PVQR is still better than BQR for the same type of queries. Because Mondial is simple and good performance is yielded by all systems ($P@k > 0.7$) and especially PVQR ($P@k > 0.9$), we omit its results for the sake of space.

The best performance achieved by PVQR in all the cases clearly reflects the superiority of PVQR and its usage of the graph data structure. The difference in performance between PVQR and other systems is most evident for IMDb. This is because IMDb contains a much

6. Query Rewriting for Keyword Search on Graphs

Query Set	BQR	PVQR/H		PVQR	
	MRR	MRR	Impr.	MRR	Impr.
Mondial(Clean)	0.88	0.97	10.2%	0.97	0.0%
Mondial(Rule)	0.88	0.97	10.2%	0.97	0.0%
Mondial(Rand)	0.78	0.98	25.6%	0.99	1.0%
IMDb(Clean)	0.38	0.74	94.7%	0.82	10.8%
IMDb(Rule)	0.32	0.74	131.3%	0.82	10.8%
IMDb(Rand)	0.19	0.68	257.9%	0.77	13.2%
Wikipedia(Clean)	0.73	0.89	21.9%	0.94	5.6%
Wikipedia(Rule)	0.71	0.85	19.7%	0.89	4.7%
Wikipedia(Rand)	0.72	0.88	22.2%	0.93	5.7%

Table 6.7.: The respective effects of our probabilistic model and additional heuristics on effectiveness of query rewriting.

larger data graph than other datasets and thus the graph structure is more crucial for finding the Steiner graphs here.

Furthermore, we investigate the respective contributions of our probabilistic model and the additional heuristics to effectiveness of query rewriting. Table 6.7 illustrates MRRs for BQR completely based on the ad-hoc heuristics, our probabilistic model (PVQR/H) without the heuristics on top and the default PVQR integrating also the additional heuristics. While the results illustrate a significant improvement achieved by PVQR/H on BQR, especially for IMDb, PVQR improves PVQR/H relatively slightly by adding heuristics. This clearly shows the benefit of using our probabilistic model to effectiveness of query rewriting. In addition, the improvement yielded by the additional heuristics witnesses the adaptability of our approach.

6.5.4. Impact on Efficiency of Keyword Search

For investigating the impact of query rewriting on keyword search, we employed two keyword search systems: the bidirectional search solution (*BDS*) (Kacholia et al., 2005) explores paths between keyword elements online, while the keyword join approach (*KJ*) (Ladwig & Tran, 2011) materializes paths in the index and only join them online. *KJ* was shown to be faster than *BDS* but also employs a larger index. Given the three query sets (Clean, Rule, Rand), we use them as they are (*NQR*), rewrite them using PVQR and BQR to obtain 9 types of queries. For queries with rewriting, we use the top-1 as input to the keyword search systems. For reasons of space, we omit the Mondial results and explicitly discuss them in the text only when they are relatively different from the other results.

Figs. 6.6(a-d) illustrate the average time for processing these 9 types of keyword queries using *KJ* (Figs. 6.6(a-b)) and *BDS* (Figs. 6.6(c-d)) for IMDb and Wikipedia. Further, the time is decomposed into query rewriting and keyword query processing components, e.g., $QR(\text{Clean})$

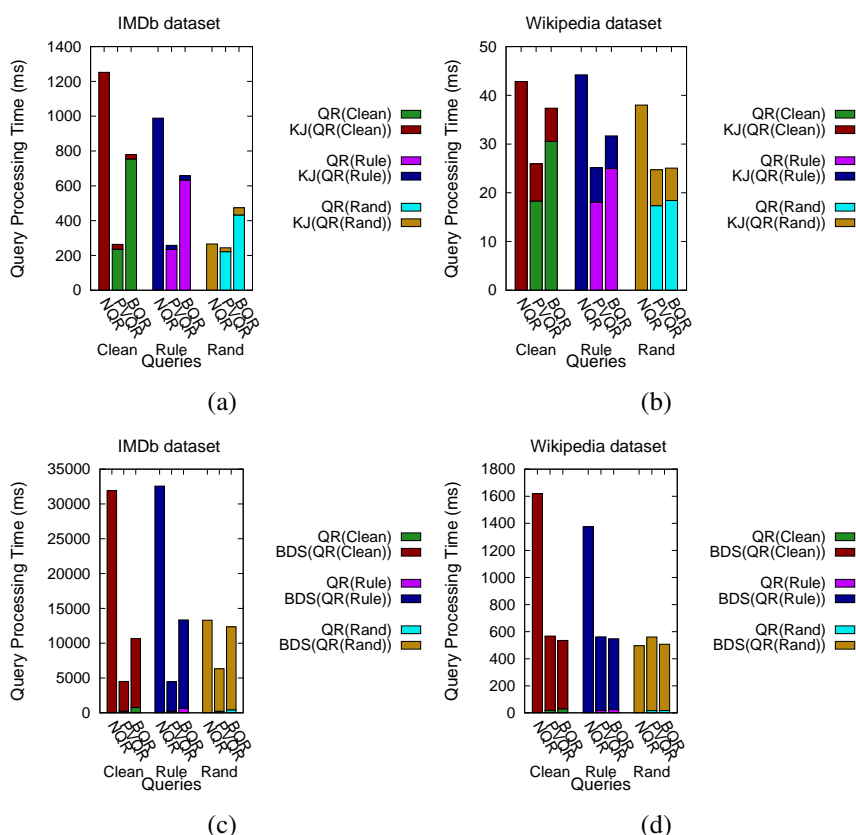


Figure 6.6.: Evaluation results for efficiency of keyword search.

is the time needed for rewriting the Clean queries, and $KJ(QR(Clean))$ is the time KJ needs to process these rewritten queries.

The ratio of these two components seems to depend on the complexity of keyword query processing (reflected in the dataset size and query length), and the systems used for that: Clearly, with a slower system (BDS), the fraction of time needed for query rewriting is smaller (compare (c+d) with (a+b)). With higher complexity (IMDb), query rewriting makes up a larger part of the total (compare (b+d) with (a+c)). Meanwhile, we also observe that *with slower systems as well as higher complexity, the positive effect of query rewriting on keyword query processing is also higher* (compare NQR with PVQR and BQR), e.g., the highest reduction in time PVQR and BQR can achieve is for BDS over IMDb. This is because for longer queries, more keywords can be grouped into segments, and with slower system and larger datasets, this *effect of segmentation* is more evident.

Clean queries take more time than Rule queries, which in turn, is more difficult to process than

Rand queries. Similar to the effect observed in the query rewriting experiment, this is due to the number of intermediate results, e.g., for Rand query keywords, keyword search systems find fewer matching elements. Accordingly, *query rewriting (PVQR, BQR) leads to reduction in time* especially for Clean and Rule queries, i.e., yields better performance than NQR. In particular, PVQR is about *5-6 times (2-3 times) faster* than NQR for IMDb (Wikipedia). For Rand query, less time is needed in total (compared to Rule and Clean). Hence, there is less room for time reduction through rewriting in this case. Also, the segmentation effect is small here as Rand queries yield fewer correct tokens that can be grouped.

We observe that *PVQR is 2-3 times faster than BQR* for IMDb and is slightly better than or similar to BQR for Wikipedia. Actually, BQR is slightly better than PVQR for many Wikipedia queries. However, this is entirely due to the fact that BQR requires less time for keyword query processing. BQR prefers rare terms, which yield fewer (relevant) keyword elements to be processed. However, the fact that BQR processes fewer (relevant) results is not shown in this experiment, but becomes evident in the following study.

6.5.5. Impact on Effectiveness of Keyword Search

Both KJ and BDS implement a combination of proximity- and TFIDF-based ranking studied in the benchmark (Coffman & Weaver, 2010). Since both systems use the same ranking, keyword search answers are very similar, hence we only show results for KJ. We use *Precision* and *Recall* for evaluating keyword search results obtained for the 9 types of queries. Given Q , let R_k be the top- k results and R^* the ground truth results captured by the benchmark. For different values of k , we have $Precision = \frac{|R^* \cap R_k|}{|R_k|}$, and $Recall = \frac{|R^* \cap R_k|}{|R^*|}$.

For different k , Figs. 6.7(a-b) plot the precision achieved by KJ for the 9 types of queries over IMDb and Wikipedia. As expected, precision consistently decreases with higher k . The queries rewritten by PVQR achieve the best results and the worst results are yielded for BQR queries. Improvements achieved by PVQR over NQR are largest for the dirty queries Rule and Rand (up to 60% for $k = 1$) and smallest for Clean (up to 10%). BQR obtains better results than NQR only for Rand queries. Thus, the conclusions are: Higher precision can be obtained for Clean queries compared to dirty queries (with or without rewriting). Rewriting with *PVQR improves precision for all types of queries while BQR yields better results only for the most dirty queries (Rand)*. Note that these results correspond to the ones from the rewriting experiments, where PVQR produces better rewrites than BQR. Hence, we conclude that *better query rewrites yield higher precision of keyword search results*.

Figs. 6.7(c-d) show that for recall, similar differences can be observed between the approaches (NQR, PVQR and BQR) and queries (Clean, Rule and Rand) for small values of k . However, while PVQR achieves highest recall for all Wikipedia queries, it performs slightly worse than NQR on Clean IMDb queries when $k \geq 10$. The conclusion is *PVQR improves recall on dirty queries but not on Clean queries* when a large number of results have to be considered.

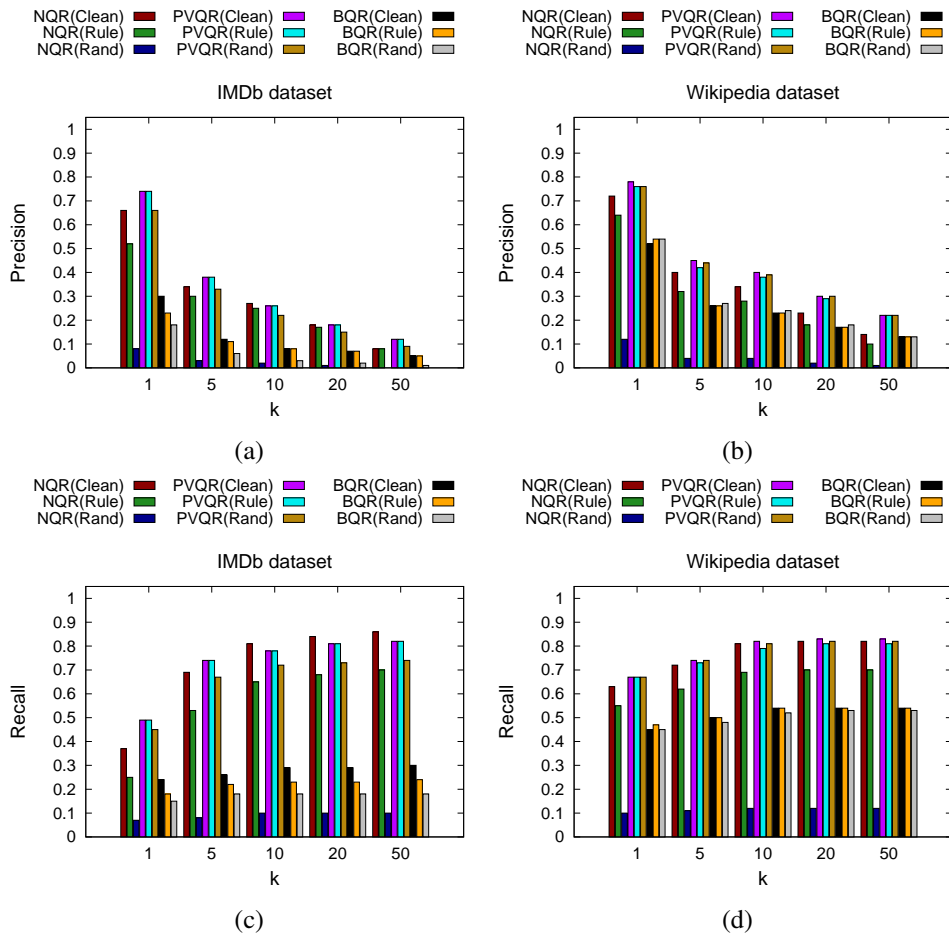


Figure 6.7.: Evaluation results for effectiveness of keyword search.

The relative differences between the approaches and between the queries are the same for the Mondial dataset. However, we note that precision and recall for Mondial are consistently higher than for IMDb and Wikipedia.

6.5.6. Analysis of Impact of Query Rewriting

In the experiments, we observe that token rewriting helps to find more relevant keyword elements and thus improve the quality of the final keyword search answers for dirty queries. This explains why PVQR achieves significantly higher precision and recall than NQR for Rule and Rand queries. BQR improves NQR only for Rand queries because it yields poor results for retrieving the top-1 query rewrite that we use as input to the keyword search systems.

Existing keyword search systems usually use a threshold to restrict the size of the retrieval list of keyword elements, where relevant ones might be excluded. Here we use the default setting in (Ladwig & Tran, 2011) to retrieve the top-300 matching elements for each keyword. In essence, query segmentation leads to fewer compound keywords. Clearly, due to the higher selectivity of compound keywords yielded by segmentation, it is more likely to have the correct keyword elements. For queries without segmentation, the retrieval list may contain no or fewer correct ones, especially for the common (non-discriminative) keywords. The observation that PVQR obtains better results than NQR even for Clean queries confirms our analysis. The only exception is the recall for Clean IMDb queries when k is large. This is because while query segmentation reduces the search space, it may not preserve all true positives, hence it cannot yield higher recall.

In terms of efficiency, query segmentation has a positive effect because fewer keywords have to be processed. This effect is evident for clean queries, where efficiency improvements can be entirely attributed to query segmentation. While token rewriting improves the quality of the keyword search, it has a negative effect on efficiency. The reason is that clean tokens yield more keyword elements that have to be processed. However, the combined effect of token rewriting and query segmentation on efficiency is still positive, as indicated by improvements obtained for the dirty query sets.

In summary, query rewriting has a clear positive effect on precision of keyword search, while still preserving high recall when the number of results is not too large. Also, it improves efficiency because the positive effect of query segmentation is larger than the negative effect of token rewriting.

6.6. Related Work

We firstly discuss the previous work that specifically targets token rewriting and query segmentation, and then the related work of query rewriting that tackles both tasks.

Token Rewriting. This problem, a.k.a. spell checking, has attracted interest in the Web context (Li et al., 2006; Cucerzan & Brill, 2004). Syntactic and semantic distances to dictionary words and the context constitute the main feature space. Based on such features, XClean (Lu et al., 2011) and our approach employ the same error model (Mays et al., 1991) to estimate the probability of token rewrite. The difference is that while XClean assumes the specific XML type semantics in a semi-structured setting which does not exist in our more general graph setting, our approach takes into account connectivity information to prune token rewrites that do not lead to valid results. Further, XClean only considers the problem of token rewriting (thus, only Sec. 6.3.2 contains overlaps with XClean).

Query Segmentation. Query segmentation is extensively studied in the Web search setting (Tan & Peng, 2008; Bergsma & Wang, 2007; Jones et al., 2006). In (Jones et al., 2006),

query segmentation is based on mutual information between pairs of query keywords. The work in (Bergsma & Wang, 2007) uses supervised learning to decide whether to create a segment boundary at each keyword position, and (Tan & Peng, 2008) proposes an unsupervised method for query segmentation using generative language models. While the use of probabilistic model is not new in the text-centric Web search setting (e.g., (Tan & Peng, 2008)), our work is different to the previous work (including (Pu & Yu, 2008) for structured data) in that we use connectivity information in the data for focusing on segments that lead to valid results.

Query Rewriting. The most related work (Pu & Yu, 2008) first introduces the problem of keyword query rewriting over the relational database. It targets both token rewriting and segmentation based on the ad-hoc heuristics. The subsequent work (Gao et al., 2011) explores query logs to improve the quality of query rewriting using the same heuristics. In contrast to the existing work, we propose a probabilistic framework to enable the query rewriting problem to be studied in a more principled way. Different from query rewriting, (Yao et al., 2012) investigates the problem of query reformation to provide totally new queries which are similar or related to the initial one.

6.7. Conclusions

In this chapter, we discuss drawbacks of existing work on query rewriting and present a principled probabilistic approach to this problem. In the experiments, we show that for query rewriting, our approach is several times faster than the state-of-the-art baseline and also yields higher quality of rewrites especially for large datasets. Most importantly, we show that these improvements also carry over to the actual keyword search. Our approach consistently improves keyword search, i.e., yields several times faster keyword search performance and substantially improves the precision and recall of keyword search results, while the baseline also provides faster performance but compromises on the quality of results, i.e., achieves good results only for very dirty queries.

Part IV.

Cross-lingual Semantics

7. Cross-lingual Linked Data Lexica

In this chapter, we introduce our cross-lingual linked data lexica, called *xLiD-Lexica*, which has been constructed by exploiting the multilingual Wikipedia and linked data sources, especially DBpedia. Firstly, we provide the reference association between entities and labels, where labels (aka surface forms) are words or phrases that can be used to refer to entities. The reference association of each pair of label and entity captures the relationship in the sense that to which extent the label refers to the corresponding entity and thus it is an intended sense of the label. Besides that, we also provide the co-occurrence association between entities and labels, where we utilize labels that co-occur with an entity in its immediate context to derive their co-occurrence frequency. Apart from labels, there are many more words contained in Wikipedia, which could be important resources for many tasks. Therefore, we also derive the co-occurrence association between entities and words. In order to derive such associations between entities and words / labels across languages, cross-language links that connect Wikipedia articles in different languages describing equivalent entities have been employed.

7.1. Introduction

With the ever-increasing quantities of knowledge published as Linked Open Data (LOD) on the Web, Natural Language Processing (NLP) technologies can both, benefit from and support such linked data repositories. For instance, NLP and its applications often involve various linked data resources, which can be utilized to assist NLP modules in a variety of tasks. On the other hand, NLP technologies can help to grow these structured data sources by automatic extraction of information from text.

DBpedia (Bizer et al., 2009b), as a large data source, stays in the center of the LOD cloud. It is a crowd-sourced community effort to extract structured information from Wikipedia in different languages and to make this information available on the Web. In recent years, DBpedia has become a valuable resource for language technologies. However, the information in DBpedia is mostly extracted from Wikipedia infoboxes, resulting in rich structured data, while the natural language texts in Wikipedia are to a large extent not exploited. The deficiency in natural language expressions for DBpedia resources hinders its more wide-spread application in NLP tasks. On the other hand, multilinguality and cross-linguality have emerged as issues of major interest nowadays. Although DBpedia is a large multilingual knowledge base (Mendes et al., 2012), the rich cross-lingual structures contained in Wikipedia are missing there.

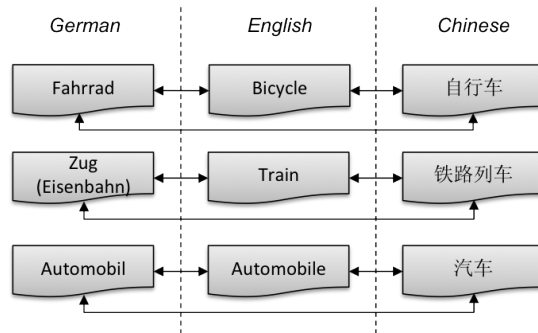


Figure 7.1.: Examples of interlingual resources in Wikipedia. The connecting arrows represent cross-language links between Wikipedia articles in different languages.

The goal of this work is to bridge the gap between cross-lingual NLP and LOD data sources, in particular DBpedia. Besides infoboxes, extracting additional information from the natural language text in Wikipedia and analyzing the semantics of its structures, such as redirect pages and anchor text of hyperlinks, would help to enrich DBpedia from the NLP perspective. Furthermore, Wikipedia currently supports approximately 280 languages and it also aligns articles in different languages that provide information about the same entity. Since a wide range of applications can benefit from its multilinguality and cross-linguality, it is essential to integrate the rich multilingual and cross-lingual information contained in Wikipedia into DBpedia, such that it is possible to leverage the huge amount of knowledge across languages.

The remainder of this chapter is structured as follows. In Sec. 7.2, we present our extraction process based on various structures in Wikipedia along with the dataset description in Sec. 7.3. Then we discuss the related work in Sec. 7.4 before we conclude in Sec. 7.5.

7.2. Methodology

In this section, we firstly introduce some useful structures in Wikipedia. Based on that, we then describe the extraction process for various associations between natural language expressions in Wikipedia and DBpedia entities. Furthermore, we use the links between DBpedia and other LOD sources to derive natural language expressions associated with their entities.

7.2.1. The Structures in Wikipedia

Wikipedia is the largest online encyclopedia up to date, which is an ever-growing useful source of manually defined information contributed by millions of users over the Web. All of Wikipedia's content is presented on pages, such as articles and categories.

Interlingual Resources. Articles supply the bulk of informative content in Wikipedia. Each article describes a single resource and they often provide information about the equivalent resources in different language versions of Wikipedia. Based on the information and structure in Wikipedia, we define the *interlingual resource* using cross-language links, which are created by adding references to corresponding articles in other languages to the source articles. An interlingual resource corresponds to Wikipedia articles in different languages which are connected by the cross-language links. As shown in Figure 7.1, the interlingual resource $\langle Bicycle \rangle$ is defined by the English article *Bicycle* and all articles that are connected to this article via cross-language links, e.g., *Fahrrad* (German) and 自行车 (Chinese). In our example, for each interlingual resource, i.e., $\langle Bicycle \rangle$, $\langle Train \rangle$ and $\langle Automobile \rangle$, there are three articles in English, German and Chinese that are fully connected across languages.

Labels. In addition, Wikipedia provides several structures that associate articles with natural language expressions (including words and phrases), also called *surface forms* or *labels* that can be used to refer to the corresponding resources. Now we introduce the following elements:

- *Title of Wikipedia article:* The title of each article is generally the most common name for the resource described in this article. For example, the English article about the resource $\langle Bicycle \rangle$ has the title “Bicycle”, and the corresponding German and Chinese articles have the titles “Fahrrad” and “自行车”, respectively.
- *Redirect page:* A redirect page exists for each alternative name, which can be used to refer to a resource in Wikipedia. For example, the articles titled “Pedal cycle” in English, “Stahlross” in German and “脚踏车” in Chinese are redirected to the articles titled “Bicycle”, “Fahrrad” and “自行车”, respectively. Thus, all these terms can be used to represent the resource $\langle Bicycle \rangle$. Redirect pages often indicate synonyms, abbreviations or other variations of the pointed resources.
- *Anchor text of hyperlinks:* The articles in Wikipedia often contain hyperlinks pointing to the pages of resources mentioned in the articles. For example, there are anchor texts “bike” appearing 50 times in English Wikipedia, “Rad” appearing 8 times in German Wikipedia and “单车” appearing 204 times in Chinese Wikipedia pointing to the articles about the resource $\langle Bicycle \rangle$. The anchor text of a link pointing to a page provides the most useful source of synonyms and other variations of the linked resource.

Words. Besides the structural elements, Wikipedia, as an extensive multilingual corpus, also provides plain text – that is, the full content of Wikipedia articles covering a wide range of topics, such as, but not limited to, arts, history, events, geography, mathematics and technology – in a vast amount with regard to the number of pages per language. Many NLP tasks can benefit from such unstructured resources, especially *words*.

7.2.2. Extraction Process

In the following, we briefly describe the DBpedia entities and then discuss the extraction process for various associations between natural language elements and DBpedia entities.

DBpedia entities. DBpedia is mainly extracted from structured information in Wikipedia editions in multiple languages. For each Wikipedia article, there exists a Uniform Resource Identifier (URI) in DBpedia (henceforth also called *DBpedia entity*). For example, the DBpedia entities `dbpedia:Bicycle`, `dbpedia-de:Fahrrad` and `dbpedia-zh:自行车`¹ correspond to the Wikipedia articles *Bicycle*, *Fahrrad* and *自行车*, respectively. Moreover, each DBpedia entity can be aligned with an interlingual resource and thus its corresponding Wikipedia articles in different languages². In the above example, each of the mentioned DBpedia entities captured in one language can be mapped to the interlingual resource $\langle Bicycle \rangle$ and thus connected with all the three Wikipedia articles in English, German and Chinese. Through an interlingual resource as a hub, all its corresponding DBpedia entities in different languages are connected. In the following, we mainly discuss the associations between natural language elements and interlingual resources, each of which represents a collection of its corresponding DBpedia entities in different languages.

Label and Resource Reference Associations. We now discuss the reference associations between labels and interlingual resources. On the one hand, labels could encode synonymy, because an interlingual resource could be represented by many labels, even in different languages. For example, the interlingual resource $\langle Bicycle \rangle$ can be denoted by the labels “bicycle” and “bike” in English, “Fahrrad” and “Rad” in German, “自行车” and “单车” in Chinese. On the other hand, labels could also encode polysemy, because a label could refer to multiple resources. For example, the label “bike” can stand for both interlingual resources $\langle Bicycle \rangle$ and $\langle Motorcycle \rangle$ and thus can represent many DBpedia entities in different languages, such as `dbpedia:Bicycle`, `dbpedia-de:Fahrrad`, `dbpedia-zh:自行车`, and `dbpedia:Motorcycle`, `dbpedia-de:Motorrad`, `dbpedia-zh:摩托车`.

Because all the labels are extracted from Wikipedia articles, the associated usage statistics can be mined for deriving the relationship between labels and interlingual resources. For example, the label “bike” refers to the resource $\langle Bicycle \rangle$ 50 times and to $\langle Motorcycle \rangle$ only 10 times such that “bike” is more likely to refer to $\langle Bicycle \rangle$. Based on the above observations, we define the probability $P(r|l)$ to model the likelihood that label l refers to resource r as

$$P(r|l) = \frac{count_{\text{link}}(r, l)}{\sum_{r_i \in R_l} count_{\text{link}}(r_i, l)} \quad (7.1)$$

¹We use prefix `dbpedia` for <http://dbpedia.org/resource/>, `dbpedia-de` for <http://de.dbpedia.org/resource/> and `dbpedia-zh` for <http://zh.dbpedia.org/resource/>.

²In this work, we use the terms *resource*, *interlingual resource* and *DBpedia resource* interchangeably, since they can be easily mapped to each other.

where $count_{\text{link}}(r, l)$ denotes the number of links using l as label pointing to r as destination and R_l is the set of resources having label l . Then, we can semantically represent each term matching a label l as a weighted vector of the resources r according to the weight $P(r|l)$.

In addition to $P(r|l)$, we also define the probability $P(l|r)$ to model the likelihood of observing l as label given resource r as

$$P(l|r) = \frac{count_{\text{link}}(r, l)}{\sum_{l_i \in L_r} count_{\text{link}}(r, l_i)} \quad (7.2)$$

where L_r is the set of labels that can refer to resource r . Given a interlingual resource r , we can therefore retrieve all terms, which are used as labels to refer to r in Wikipedia, together with the weights $P(l|r)$.

To calculate the strength w.r.t. the reference association of a pair of label l and resource r , the probabilities $P(r|l)$ and $P(l|r)$ are further processed to generate the point-wise mutual information (PMI) of l and r , defined as

$$PMI(l, r) = \log \frac{P(l, r)}{P(l)P(r)} = \log \frac{P(l|r)}{P(l)} = \log \frac{P(r|l)}{P(r)} \quad (7.3)$$

We define the probability $P(l)$ that label l appears as links in Wikipedia, no matter which resources it refers to, as

$$P(l) = \frac{\sum_{r_i \in R_l} count_{\text{link}}(r_i, l)}{N_{\text{link}}} \quad (7.4)$$

where N_{link} denotes the total number of links in Wikipedia. Similarly, we define the probability $P(r)$ that resource r is linked in Wikipedia regardless of the used label, as

$$P(r) = \frac{\sum_{l_i \in L_r} count_{\text{link}}(r, l_i)}{N_{\text{link}}} \quad (7.5)$$

According to Equation 7.3, 7.4 and 7.5, we derive the strength w.r.t. reference association of a pair of label l and resource r as follows

$$PMI(l, r) = \log \frac{count_{\text{link}}(r, l) \times N_{\text{link}}}{\sum_{r_i \in R_l} count_{\text{link}}(r_i, l) \times \sum_{l_i \in L_r} count_{\text{link}}(r, l_i)} \quad (7.6)$$

In terms of $P(r|l)$, $P(l|r)$ and $PMI(l, r)$, it is observed that the main difference between them lies in the normalization factor in the denominator of Equation 7.1, 7.2 and 7.6, respectively. Two terms, namely $\sum_{r_i \in R_l} count_{\text{link}}(r_i, l)$ standing for the frequency that label l appears as links and $\sum_{l_i \in L_r} count_{\text{link}}(r, l_i)$ denoting the frequency that resource r is linked in Wikipedia, are involved. The normalization factor of $P(r|l)$, i.e., $\sum_{r_i \in R_l} count_{\text{link}}(r_i, l)$, does not affect the ranking of r when l is specified (since the probabilities for different r are divided by the

same term). Similarly, the normalization factor of $P(l|r)$, i.e., $\sum_{l_i \in L_r} count_{link}(r, l_i)$, does not affect the ranking of l when r is specified.

The normalization factor of $PMI(l, r)$ differs from $P(r|l)$ and $P(l|r)$ by involving both terms. Intuitively, it correlates to the inverse of frequency that label l and resource r are used as links in Wikipedia. It means that labels and resources more rarely linked give higher contribution to the total association strength, which is similar to the inverse document frequency (IDF) widely used in the IR area. In this way, $PMI(l, r)$ attempts to make the association strength for different pairs of l and r comparable based on not only the correlation between l and r (represented by $count_{link}(r, l)$) but also their individual discriminability (represented by $\sum_{r_i \in R_l} count_{link}(r_i, l)$ and $\sum_{l_i \in L_r} count_{link}(r, l_i)$). Based on this guide, we can choose among $P(r|l)$, $P(r|l)$ and $PMI(l, r)$ for the particular tasks at hand.

Label and Resource Co-occurrence Associations. The reference association of a pair of label l and resource r captures the relationship in the sense that to which extent l refers to r and thus r is an intended sense of l . Besides that, we also utilize labels that frequently co-occur with a resource in its immediate context to derive co-occurrence associations.

The link structure in Wikipedia allows us to determine the labels within the context of a resource (defined by a sliding window of k sentences). To illustrate, let us consider the following paragraphs extracted from three Wikipedia articles in English, German and Chinese, which are all related to the resource $\langle Bicycle \rangle$.

Example 6. *Cycling.* Cycling, also called bicycling or biking, is the use of bicycles for transport, recreation, or for sport. Persons engaged in cycling are referred to as “cyclists”, “bikers”, or less commonly, as “bicyclists”. Apart from two-wheeled bicycles, “cycling” also includes the riding of unicycles, tricycles, quadracycles, and similar human-powered vehicles (HPVs).

Example 7. *Fahrradfahren.* Der Ausdruck Fahrradfahren, auch Radfahren, bezeichnet die Fortbewegung auf einem Fahrrad. Er bezeichnet auch die Sportart Fahrradfahren, die als Freizeitbeschäftigung oder als sportlicher Wettkampf bis hin zum Leistungssport betrieben wird.

Example 8. *自行车运动.* 自行车运动常指借助自行车（或称单车）开展的各种运动的总称，属于借助人力推动的半机械化运动，极少使用单轮车、三轮车、四轮车或其他用于运输、娱乐或运动的人力车辆开展此项运动。自行车运动在公路或小道上进行，根据不同的环境和要求开展此项活动，如自行车旅行、越野自行车运动、雪地自行车运动等等。

All the underlined words and phrases represent labels on the one hand, and represent links to the corresponding Wikipedia articles and thus the aligned resources on the other hand. In this way, each label can be semantically interpreted as a weighted vector of its neighboring linked resources and each resource can be treated as a weighted vector of its neighboring labels

	English	German	Spanish	Catalan	Slovenian	Chinese
#Resources	4.2M	1.4M	1.0M	0.4M	0.1M	0.7M
#Labels	13M	4.6M	3.2M	0.9M	0.3M	1.3M
#Words	2.6B	908M	666M	224M	48M	321M

Table 7.1.: Statistics about words and labels in Wikipedia.

in different languages. For example, the label *human-powered vehicles* can be represented as a vector of the interlingual resources $\langle Bicycle \rangle$, $\langle Transport \rangle$, $\langle Recreation \rangle$, $\langle Sport \rangle$, $\langle Unicycle \rangle$, $\langle Tricycle \rangle$ and $\langle Quadracycle \rangle$ and thus also as a vector of corresponding DBpedia entities captured in any supported languages. And the interlingual resource $\langle Bicycle \rangle$ and each corresponding DBpedia entity captured in one language can be represented as a vector of the labels, such as *transport*, *recreation*, *sport*, *unicycles*, *tricycles*, *quadracycles* and *human-powered vehicles* in English, *Sportart Fahrradfahren* and *Leistungssport* in German, 运动, 单轮车, 三轮车 and 四轮车 in Chinese.

Next, we discuss the weights of the resources as interpretations of a label. For this, we define the probability $P_k(r|l)$ to model the likelihood that given a label l , the resource r co-occur with it in a window of k sentences as

$$P_k(r|l) = \frac{\text{count}_{\text{co-occur}}(r, l)}{\sum_{r_i \in R_l} \text{count}_{\text{co-occur}}(r_i, l)} \quad (7.7)$$

where $\text{count}_{\text{co-occur}}(r, l)$ denotes the frequency that l and r co-occur in a window of k sentences and R_l is the set of resources that co-occur with label l .

Then, we discuss the weights of the relevant labels given a resource. For this, we define the probability $P_k(l|r)$ to model the likelihood of l appearing in the context of resource r with size k as

$$P_k(l|r) = \frac{\text{count}_{\text{co-occur}}(r, l)}{\sum_{l_i \in L_r} \text{count}_{\text{co-occur}}(r, l_i)} \quad (7.8)$$

where L_r is the set of labels that co-occur with resource r .

Similarly, we calculate the strength w.r.t. the co-occurrence association of a pair of label l and resource r based on $P_k(r|l)$ and $P_k(l|r)$ as

$$PMI_k(l, r) = \log \frac{\text{count}_{\text{co-occur}}(r, l) \times N_{\text{co-occur}}^{\text{label}}}{\sum_{r_i \in R_l} \text{count}_{\text{co-occur}}(r_i, l) \times \sum_{l_i \in L_r} \text{count}_{\text{co-occur}}(r, l_i)} \quad (7.9)$$

where $N_{\text{co-occur}}^{\text{label}}$ is the sum of the frequency that label l and resource r co-occur in a window of k sentences for all pairs of l and r . The difference between $P_k(r|l)$, $P_k(l|r)$ and $PMI_k(l, r)$ w.r.t. co-occurrence association between labels and resources is similar to the reference association as discussed before.

Word and Resource Co-occurrence Associations. Apart from labels, there are many more words contained in Wikipedia for different languages, which can play an important role in NLP. The biggest advantage of word-based NLP approaches is the large amount of available data, such that they are not subject to data sparsity issues. As shown in Table 7.1, the number of words in Wikipedia significantly exceeds the number of extracted labels for the different languages. For example, the English Wikipedia alone contains over 2.6 billion words, over 100 times as many as the next largest English-language encyclopedia, Encyclopaedia Britannica, while it has only 13 million labels. Therefore, it is also crucial to derive the co-occurrence association between words and resources.

First, we define the probability $P_k(r|w)$ to model the likelihood that given a word w , the resource r co-occur with it in a window of k sentences as

$$P_k(r|w) = \frac{\text{count}_{\text{co-occur}}(r, w)}{\sum_{r_i \in R_w} \text{count}_{\text{co-occur}}(r_i, w)} \quad (7.10)$$

where $\text{count}_{\text{co-occur}}(r, w)$ denotes the frequency that word w and resource r co-occur in a window of k sentences and R_w is the set of resources that co-occur with word w . For each word w in one language, we can derive a vector of weighted co-occurred resources r with the weight $P_k(r|w)$. In Example 6, the word *bicycling* can be represented as a weighted vector of the interlingual resources $\langle \text{Bicycle} \rangle$, $\langle \text{Transport} \rangle$, $\langle \text{Recreation} \rangle$, $\langle \text{Sport} \rangle$, $\langle \text{Unicycle} \rangle$, $\langle \text{Tricycle} \rangle$, $\langle \text{Quadracycle} \rangle$ and $\langle \text{Human-powered transport} \rangle$ and also the corresponding DBpedia entities captured in any supported languages.

Next, we define the probability $P_k(w|r)$ to model the likelihood of word w appearing in the context of resource r with size k as

$$P_k(w|r) = \frac{\text{count}_{\text{co-occur}}(r, w)}{\sum_{w_i \in W_r} \text{count}_{\text{co-occur}}(r, w_i)} \quad (7.11)$$

where W_r is the set of words that co-occur with resource r . For each resource, a vector of words w appearing in the context of r with weights $P_k(w|r)$ can be generated. In the previous examples, regarding the resource $\langle \text{Bicycle} \rangle$ and all corresponding DBpedia entities captured in different languages, we can generate a vector of the words, such as *cycling* and *cyclists* in English, *Radfahren* and *Freizeitbeschäftigung*, in German, 半机械化和 运输 in Chinese.

Finally, we calculate the strength w.r.t. the co-occurrence association of a pair of word w and resource r based on $P_k(r|w)$ and $P_k(w|r)$ as

$$PMI_k(w, r) = \log \frac{\text{count}_{\text{co-occur}}(r, w) \times N_{\text{co-occur}}^{\text{word}}}{\sum_{r_i \in R_l} \text{count}_{\text{co-occur}}(r_i, w) \times \sum_{w_i \in W_r} \text{count}_{\text{co-occur}}(r, w_i)} \quad (7.12)$$

where $N_{\text{co-occur}}^{\text{word}}$ is the sum of the frequency that word w and resource r co-occur in a window of k sentences for all pairs of w and r . We omit the discussion about the difference between $P_k(r|w)$, $P_k(r|w)$ and $PMI_k(w, r)$, because it is similar to that between $P(r|l)$, $P(r|l)$ and $PMI(l, r)$ as discussed before.

	Our Datasets			DBpedia NLP Datasets	
	<i>#Label Resource Reference Associations</i>	<i>#Label Resource Co-occurrence Associations</i>	<i>#Word Resource Co-occurrence Associations</i>	<i>#DBpedia Lexicalizations Entries</i>	<i>#DBpedia Topic Signatures Entries</i>
English	15,237,596	104,560,077	313,266,917	2,176,869	8,438,400
German	5,342,851	42,316,145	172,033,719	–	–
Spanish	3,563,379	34,404,641	106,951,335	–	–
Catalan	1,022,815	8,161,564	29,753,250	–	–
Slovenian	380,522	26,638,003	25,249,677	–	–
Chinese	1,425,827	16,286,187	19,851,666	–	–
<i>Total</i>	26,972,990	232,366,617	667,106,564	2,176,869	8,438,400

Table 7.2.: Statistics of our datasets and DBpedia NLP Datasets.

Our Datasets	English	German	Chinese	DBpedia NLP Datasets	English
<i>Label Resource Reference</i>	Michael Jordan Jordan Air Jordan His Airness MJ23	Michael Jordan Jordan Air Jordan His Airness Jordan, Michael	迈克尔·乔丹 麥可·喬丹 麥可·喬登 米高·佐敦 邁克爾·喬丹	<i>DBpedia Lexicalizations</i>	Michael Jordan Jordan MJ – –
<i>Label Resource Co-occurrence</i>	Scottie Pippen Dennis Rodman Chicago Bulls United Center NBA	Chicago Bulls NBA Basketball Scottie Pippen San Antonio Spurs	波士頓人 洛杉磯湖人 城76人 芝加哥公牛 聖安東尼奧馬刺		
<i>Word Resource Co-occurrence</i>	nba basketball bulls chicago game	bulls chicago spieler nba basketballspieler	洛杉磯 凱爾特人 波士頓 芝加哥 薩克拉門托	<i>DBpedia Topic Signatures</i>	game nba team – –

Table 7.3.: Examples of top-5 results from our datasets and DBpedia NLP datasets for English DBpedia entity dbpedia:Michael_Jordan.

7.3. Datasets

In this section, we describe our datasets extracted based on the methodology presented in Sec. 7.2, where we used the Wikipedia dumps of July 2013 in English, German, Spanish, Catalan, Slovenian and Chinese.

Table 7.2 provides the main statistics of our datasets w.r.t. the three associations, namely label resource reference, label resource co-occurrence and word resource co-occurrence associations. In order to compare our datasets with the most related work, Table 7.2 also provides the

statistics of DBpedia NLP Datasets³, where the Lexicalization dataset contains the information similar to our label and resource reference associations. In the Topic Signatures dataset each DBpedia resource is represented by a term vector (of size 3 in most cases) extracted from Wikipedia article content using TF-IDF weights (Mendes et al., 2012). It is observed that our datasets contain more entries than the DBpedia NLP Datasets and provide information in more languages. Table 7.3 shows the top-5 results of different associations for the English DBpedia entity `dbpedia:Michael_Jordan` from our datasets and the DBpedia NLP datasets. This conveys the impression that we achieve more comprehensive results in terms of quantity and quality compared with the DBpedia NLP datasets.

Dataset Dumps. The first version of our datasets is available⁴ as plain text files in JSON format. These files consist of a list of records, each identifying an association between a natural language expression and a resource. An example of label resource reference association between the label *MJ23* and the resource *Michael Jordan* is shown as follows:

```
{
  "id": ObjectId("53f0cfdfe4b0e7085cf241a1"),
  "label": "MJ23",
  "resource": "Michael Jordan",
  "P(r|l)": "1",
  "P(l|r)": "0.0007199424046076314",
  "PMI(l,r)": "11.102683968056724"
}
```

Accessing API and GUI. In order to effectively access and automatically embed our datasets within applications, we developed a Java API, based on MongoDB⁵ as backend, such that the dataset dumps in JSON format can be easily imported into MongoDB. The API provides a variety of methods to access different kinds of information, namely (1) the reference and co-occurrence associations with labels and words given a resource; (2) the reference and co-occurrence associations with resources given a label; (3) the co-occurrence associations with resources given a word. In addition, we ship the API with a graphical user interface (GUI) that allows the user to browse our datasets. The accessing API and the GUI are accessible as open source on GitHub⁶.

7.4. Related Work

In this section, we review the related work and discuss our contributions from two perspectives, namely dictionary datasets and lexical knowledge bases.

³<http://wiki.dbpedia.org/Datasets>

⁴<http://km.aifb.kit.edu/sites/xlid-lexica/>

⁵<http://www.mongodb.org/>

⁶<https://github.com/beyondlei/nlp-lexica>

7.4.1. Dictionary Datasets

Dictionaries contain associations that map labels to DBpedia resources as their senses, which can be applied to many applications, such as Named Entity Disambiguation (Steinmetz et al., 2013). Now we will discuss some dictionaries in the following.

The work closest to ours is the DBpedia NLP datasets (Mendes et al., 2012), which describe a number of extended resources for DBpedia that specifically aim at supporting computational linguistics tasks, where the Lexicalizations dataset contains the information similar to that captured by our label and resource reference association. Overall, there are 2 million entries of English labels and resources in the dictionary, where for each label-resource pair, the probabilities $P(r|l)$, $P(l|r)$ and the pointwise mutual information $PMI(l, r)$ are given.

The Crosswikis dictionary (Spitkovsky & Chang, 2012) is a similar, but much larger dataset for English Wikipedia concepts. It has been built at web scale and includes 378 million entries. Similar to the DBpedia Lexicalizations dataset, the probabilities $P(r|l)$ and $P(l|r)$ have been calculated and is available in the dictionary.

The means relation of YAGO⁷ has been used as dictionary by AIDA (Hoffart et al., 2011), a tool for disambiguation of named entities in text, to identify candidate entities for a (possible ambiguous) mention. The entries in the dictionary were extracted from link anchors, disambiguation pages and redirection links in Wikipedia.

Similar to the YAGO means relation, the Redirect Disambiguation Mapping (RDM) dictionary has been constructed by solving disambiguation pages and redirects and using these alternative labels additionally to the original labels of the DBpedia entities. This dictionary has been compared with other datasets in the context of Named Entity Disambiguation tasks in (Steinmetz et al., 2013).

While the use of Wikipedia for extracting reference associations between labels and DBpedia resources is not new, our work is different in that besides reference associations we also study the co-occurrence associations between different NLP elements, namely labels and words, and DBpedia resources. In addition, we provide both reference and co-occurrence associations in the cross-lingual setting by extracting labels and words from Wikipedia editions in multiple languages and exploiting cross-lingual structures of Wikipedia.

7.4.2. Lexical Knowledge Bases

In recent years, there has been a growing interest in extracting knowledge from Wikipedia and other knowledge sources such as WordNet, for constructing multilingual lexical knowledge bases. In the following, we introduce several state-of-the-art lexical knowledge bases.

⁷<http://www.yago-knowledge.org/>

WikiNet (Nastase et al., 2010) is a multilingual semantic network constructed from Wikipedia and includes semantic relations between Wikipedia entities, which are collected from the category structure, infoboxes and article contents.

UWN (de Melo & Weikum, 2009) is an automatically constructed multilingual lexical knowledge base, which is bootstrapped from WordNet and built by collecting evidence extracted from existing wordnets, translation dictionaries and parallel corpora. This results in over 800,000 words in over 200 languages in a semantic network with over 1.5 million links from words to word senses. Its extension MENTA (de Melo & Weikum, 2010) adds a large scale hierarchical taxonomy containing 5.4 million named entities and their classes, which is also built from WordNet and Wikipedia.

Similarly to UWN and MENTA, BabelNet (Navigli & Ponzetto, 2012) integrates lexicographic and encyclopedic knowledge from WordNet and Wikipedia into a unified, wide-coverage, multilingual lexical knowledge base through a novel mapping algorithm that can establish the mappings between a multilingual encyclopedic knowledge repository (Wikipedia) and a computational lexicon of English (WordNet) with high accuracy. In general, BabelNet is a multilingual semantic network, which connects concepts and named entities in a very large network of semantic relations, made up of more than 9 million entries, called Babel synsets. Each Babel synset represents a given meaning and contains all the synonyms which express that meaning in a range of different languages.

These lexical knowledge bases go one step beyond the dictionary datasets by integrating semi-structured information from Wikipedia with the relational structure of other knowledge sources into a semantic network to provide the meanings of words and phrases and to show how such meanings are semantically related based on their semantic relations. In addition to the senses, our co-occurrence associations provide complementary information about the relatedness of words and labels with DBpedia resources in a multilingual and cross-lingual setting. In this way, each word or label in any language can be represented as a vector of DBpedia resources and vice versa, which can be applied to many applications such as cross-lingual semantic relatedness (Hassan & Mihalcea, 2011).

7.5. Conclusions

In this chapter, we presented *xLiD-Lexica*, a cross-lingual linked data lexica that aims at bridging the gap between cross-lingual NLP and linked data resources, especially in DBpedia. In order to achieve this, we exploited various kinds of structures in Wikipedia, such as anchor text of hyperlinks and cross-language links, to derive different associations between natural language expressions extracted from Wikipedia editions in multiple languages and linked data resources. We believe that the extracted datasets can help to support many cross-lingual applications, such as cross-lingual semantic annotation and search.

8. Cross-lingual Keyword Query Interpretation

As a simple and intuitive way of specifying information needs, keyword queries enjoy widespread usage, but suffer from the challenges including *ambiguity* and *incompleteness*. In addition, there is an impending need for technologies that can enable *cross-lingual* information access. In this chapter, we present a knowledge base approach to cross-lingual keyword query interpretation by transforming keyword queries in different languages to their semantic representation, which can facilitate query disambiguation and expansion, and also bridge language barriers. The experimental results show that our approach achieves both high efficiency and effectiveness and considerably outperforms the baselines.

8.1. Introduction

With the ever-increasing quantities of entities in large knowledge bases (KBs) on the Web, many research activities involving *entities* have emerged recently, such as entity tagging/extraction from texts and entity linking/disambiguation with KBs. Furthermore, there is an increasing portion of Web search queries involving entities. For example, through query log analysis, Pound et al. (Pound et al., 2010) found that more than half of Web queries are related to entities. In this regard, the exploitation of *entities and their relations* in information retrieval (IR) research beyond the term-based paradigm has become an area of particular interest.

On the other hand, *multilingual* and *cross-lingual* access to information has drawn increasing attention. Nowadays, more and more people from different countries are connecting to the Internet and many Web users are able to understand more than one language. While the diversity of languages on the Web has been growing in recent years, for most people there is still very little content in their native language. As a consequence of the ability to understand more than one language, users are also interested in Web content in other languages.

In addition, keyword search has proven to be a simple and intuitive paradigm for expressing information needs of users. However, traditional keyword search systems mainly suffer from the following challenges.

Ambiguity. Keyword queries are naturally ambiguous due to the fact that keywords could refer to different things in different contexts. In the multilingual and cross-lingual settings, this

problem is more serious, e.g., “*WM*” could refer to the entity *Windows_Mobile* in English and *FIFA_World_Cup* in German¹.

Incompleteness. Keyword queries are often incomplete in the sense that instead of the full entity names, only the aliases, acronyms and misspellings are usually given in the queries. In addition, keyword queries might contain concept names representing a set of entities, e.g., “*Internet companies of China*”.

Cross-linguality. Multilingual users probably formulate their information needs using native language. However, they are interested in relevant information in any language that they can understand. In some other cases, multilingual users could issue queries consisting of keywords in multiple languages. For example, Chinese users might represent a foreign company using its original name and a local company using its Chinese name, such as “*Google 百度*” with the aim of finding the relationship between Google and Baidu, the largest search engines for English and Chinese, respectively. In addition, specifying the query language should not be the burden of users, which poses new challenges since existing techniques for language detection, such as the well-known character n-gram probability language model, do not work well for short keyword queries (Baldwin & Lui, 2010).

In order to address these challenges, we present a knowledge base approach to cross-lingual keyword query interpretation. The goal is to find entity graphs in the KB matching the keyword query, called *query entity graphs* (QEG), which reflect different semantic interpretations of the keyword query. More specifically, our approach aims to eliminate the ambiguity of keyword queries by exploiting the semantic graph of the KB to generate the top-k QEGs. It supports keyword queries matching entities in their incomplete forms, such as aliases, acronyms and misspellings instead of the full names. In addition, the matching concepts in keyword queries are automatically expanded into sets of associated entities. To the best of our knowledge, this is the first work that allows users to issue keyword queries in any language, which can even contain keywords in multiple languages, for finding the query interpretations grounded in any other languages.

It is noteworthy that this work has been incorporated into XKnowSearch!², a novel system to entity-based cross-lingual information retrieval (IR) (Zhang et al., 2016a). With the help of the resulting QEGs, XKnowSearch! allows users to further explore entity relations to refine the queries. For bridging the language barriers between queries and documents, XKnowSearch! leverages the cross-lingual query interpretation technique in this chapter and a cross-lingual semantic annotation system (Zhang & Rettinger, 2014) to construct a semantic representation of keyword queries and documents in different languages, which are then used for document retrieval. More details about XKnowSearch! will be discussed in Chapter 9.

The main contributions of this chapter are: (1) the introduction of a *knowledge base approach to cross-lingual query interpretation* by representing information needs of users as en-

¹*WM* is the abbreviation of *Weltmeisterschaft* in German, which means *World Cup*.

²<http://km.aifb.kit.edu/sites/XKnowSearch/>

tity graphs to *address the challenges* of traditional keyword search; (2) a *scoring mechanism* for *effective query interpretation ranking* by exploiting various structures in the multilingual KB; (3) a new *top-k query graph exploration algorithm* aimed for *efficient query interpretation generation*; and (4) a *separate evaluation* of the ranking mechanism and the top-k graph exploration algorithm to show that both of them lead to a *considerable improvement* over the baseline methods on *effectiveness and efficiency*, respectively.

The rest of this chapter is organized as follows. We firstly introduce the problem and provide an overview of our approach in Sec. 8.2. Details on the scoring mechanism and the top-k query graph exploration algorithm are then presented in Sec. 8.3 and Sec. 8.4, respectively. Experimental results are presented in Sec. 8.5. Finally, we survey the related work in Sec. 8.6 and conclude in Sec. 8.7.

8.2. Overview

We deal with the scenarios where queries formulated by users are sets of keywords in any language or even in multiple languages, which are unknown in advance. Given such queries, we first introduce the concepts of *key term* and *key term set* and then define the *query entity graph* (QEG) as the interpretation of a query.

Definition 15 (Key Term and Key Term Set). *Given a query Q consisting of a sequence of keywords $\langle k_1, \dots, k_n \rangle$, a key term $t = \langle k_i, \dots, k_j \rangle$ is a subsequence of Q with the start index $start(t) = i$ and the end index $end(t) = j$, for which at least one matching entity or concept can be found in the knowledge base. A key term set $T = \{t_1, \dots, t_m\}$ is a set of non-overlapping key terms resulting from Q such that for any t and t' in T either $start(t) \geq end(t')$ or $end(t) \leq start(t')$.*

For example, the keywords “*online companies of US*” could result in many key terms like *online*, *companies*, *online companies*, *US* and *online companies of US*, which could lead to different key term sets, such as $\{online, companies, US\}$ and $\{online\}$ and $\{online\}$. The key terms like *online* and *US* could refer to the entities `Online_game` and `United_States`, respectively, while *online companies of US* might refer to the concept `Internet_companies_of_the_United_States`, which has a list of associated entities belonging to it, such as `Google`, `Yahoo!` and `EBay`.

We consider the KB as a directed graph $G_{KB}(N, E)$, where each node $n \in N$ represents an entity and each edge $e(n_i, n_j) \in E$ denotes the relation between entities n_i and n_j . Given the key term sets resulting from a keyword query Q , the query interpretation of Q , i.e., the query entity graph, is defined as follows:

Definition 16 (Query Entity Graph). *A query entity graph (QEG) to a keyword query Q , denoted by $G_Q = (N_Q, E_Q)$, is a subgraph of $G_{KB}(N, E)$, which satisfies the following*

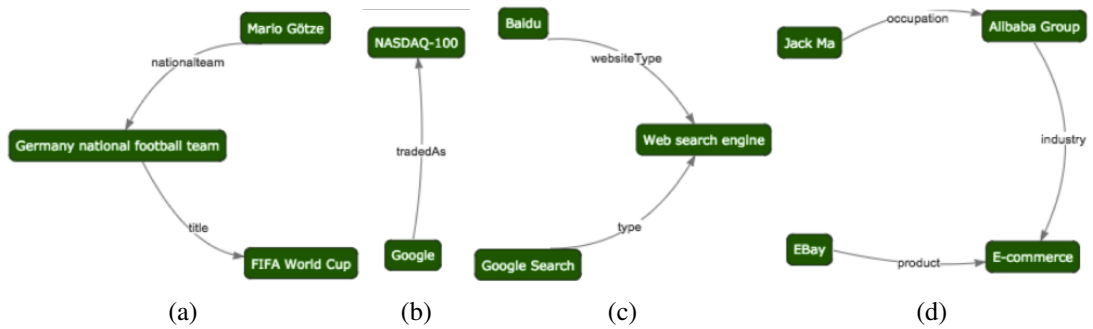


Figure 8.1.: Example QEGs generated by our system for the queries (a) “WM Götze”, (b) “online companies of US NDX”, (c) “Google 百度” and (d) “eBay 马云”.

conditions: (1) there exists at least one key term set T and for each key term $t \in T$ there is at least one entity $n_t \in N$ that matches t . The set of matching entities containing one for every $t \in T$ is $N_T \subseteq N_Q$; (2) for every possible pair $n_i, n_j \in N_T$ and $n_i \neq n_j$, there is a path $n_i \rightsquigarrow n_j$, i.e., an edge $e(n_i, n_j) \in E$ or a sequence of edges $e(n_i, n_k) \dots e(n_l, n_j)$ in E , such that every $n_i \in N_T$ is connected to every other $n_j \in N_T$.

Problem. We are concerned with the computation of QEGs from keywords in any language or even in multiple languages. Given a query Q , the goal is to find the top- k ranked QEGs, where the ranking is produced by the application of a scoring function $S : G_Q \rightarrow s$. For any given QEG G_Q , S assigns a score s that captures the degree to which G_Q matches the information need of users.

Some examples of the top-ranked QEGs generated by our system for different queries are shown in Fig. 8.1. To avoid the users’ burden of specifying the query languages, our approach does not assume any input language given by users for all the queries. In the query “WM Götze”, the keyword “WM”, which could refer to 212 entities in German and 11 entities in English, has been disambiguated as FIFA_World_Cup based on the relation to Mario_Götze. Regarding the query “online companies of US NDX”, the alias “online companies of US” referring to the concept Internet_companies_of_the_United_States has been resolved to the entity Google, which is listed in NASDAQ-100 referred to by the acronym “NDX”. For the multilingual queries “Google 百度” and “eBay 马云”, our approach can deal with them by supporting query keywords in multiple languages.

In the following, we provide an overview of the off-line preprocessing and online computation required in our approach to cross-lingual query interpretation.

Preprocessing. In this work, we use DBpedia as the knowledge base, which is a crowd-sourced community effort to extract structured information from Wikipedia in different languages. In the following, we briefly introduce the offline cross-lingual grounding extrac-

tion, where we construct the cross-lingual lexica³ by exploiting the multilingual Wikipedia to extract the cross-lingual groundings of DBpedia entities and concepts, which correspond to Wikipedia articles and categories, respectively. As Wikipedia provides several useful structures, such as titles of pages, redirect pages, disambiguation pages and link anchors, which associate entities and concepts in DBpedia with terms including words and phrases, also called *labels or surface forms*, all of them can be used to refer to the corresponding resources. In addition, Wikipedia pages in different languages that provide information about the equivalent resources are often connected through the cross-language links. Based on the above sources, for each DBpedia entity or concept grounded in one language we extract its possible surface forms in different languages. More details can be found in our previous work (Zhang et al., 2014a; Zhang et al., 2014b). The cross-lingual lexica and the knowledge extracted from DBpedia are indexed for online computation. Based on such indexed data, we are concerned with ranking the query interpretations effectively and propose a scoring mechanism for it, which will be discussed in Sec. 8.3.

Query Interpretation Computation. In order to compute the QEGs as query interpretations for a keyword query Q , all the key terms are first extracted from Q based on the cross-lingual lexica, which is also used for finding the matching entities n_t for each key term t , where either t can be used to refer to n_t directly or n_t belongs to a concept that can be referred to by t . Such key terms then result in different key term sets, each of which reflects one possible information need of users. For each key term set T and all the matching entities of its key terms, the exploration of the knowledge graph G_{KB} starts from each matching entity n_t of a key term $t \in T$ to find a connecting element, denoted by n_c , namely an entity that connects at least one starting entity n_t for all $t \in T$. Once a connecting element n_c is found, a QEG can be constructed from a set of paths that start at each n_t and meet at n_c . This process of exploration continues until the top- k QEGs have been achieved. In this chapter, we are concerned with performing this query interpretation computation efficiently and propose a new top- k graph exploration algorithm, which will be discussed in Sec. 8.4.

8.3. Query Graph Scoring

A keyword query could result in many QEGs all corresponding to possible query interpretations. This section introduces a scoring mechanism that aims to assess the relevance of QEGs for *effective query interpretation ranking*.

8.3.1. Key Term Set Score

Our approach supports query keywords in multiple languages and we assume that the languages of keywords in a query Q are unknown, such that key terms extracted from Q could

³<http://km.aifb.kit.edu/sites/xlid-lexica/>

be entity/concept names in any language. Therefore, for each language L , we define the probability $P(t_L)$ that the key term t in L , denoted by t_L^4 , is an entity name or a concept name as

$$P(t_L) = \frac{\text{count}_{\text{anchor}}(t_L)}{\text{count}_{\text{anchor}}(t_L) + \text{count}_{\text{raw}}(t_L)} \quad (8.1)$$

where $\text{count}_{\text{anchor}}(t_L)$ denotes the number of links using t as anchor text and $\text{count}_{\text{raw}}(t_L)$ denotes the frequency of t mentioned in plain text without links in Wikipedia of language L . This estimation is further smoothed by the Laplace smoothing method for the zero probability problem. As the languages of query keywords are not specified, we define the probability $P(t)$ that the key term t refers to an entity or a concept for a set of supported languages \mathcal{L} as

$$P(t) = \max_{L \in \mathcal{L}} P(t_L) \quad (8.2)$$

All the possible key terms might result in many key term sets that reflect different information needs. Therefore, we define the score of each key term set in the following. Given a keyword query Q , for each resulting key term set T , we take into account both its *importance* and *informativeness*. In general, the more often a key term t is selected as anchor text for the corresponding resources, i.e., t has larger $P(t)$, the more likely that t is important. In addition, the more keywords in Q are covered by all key terms $t \in T$, the more likely that T is informative, since it can reflect more aspects of the initial keyword query. Based on the above observation, we calculate the score of T as

$$S(T) = \frac{\sum_{t \in T} P(t) \cdot \sum_{t \in T} |t|}{|T|} \quad (8.3)$$

where $|t|$ is the number of keywords in t and $|T|$ is the number of key terms in T . While $\sum_{t \in T} P(t)$ reflects the *importance* of T , $\sum_{t \in T} |t|$ captures its *informativeness*. The denominator $|T|$ is a normalization factor used to reduce the advantage of T with more key terms. For example, $\{\text{online}, \text{companies}, \text{US}\}$ might result in a larger numerator compared with $\{\text{online companies of US}\}$.

8.3.2. Entity Matching Score

For each key term t , there might be many entities that can be referred to by t . Assuming that t is in language L , denoted by t_L , we define the probability $P(n_{L'}|t_L)$ that t_L refers to the entity $n_{L'}$ grounded in the target language L' as

$$P(n_{L'}|t_L) = \frac{\text{count}_{\text{link}}(n_L, t_L) \cdot \tau(n_L, n_{L'})}{\sum_{n_L \in N_L} \text{count}_{\text{link}}(n_L, t_L)} \quad (8.4)$$

⁴We use t for a term whose language is not observed and t_L for the same term t whose language is considered as L .

where $count_{link}(n_L, t_L)$ denotes the number of links using t_L as anchor text pointing to n_L in Wikipedia of language L and N_L is the set of entities that have name t_L . The language mapping function $\tau(n_L, n_{L'})$ is defined as

$$\tau(n_L, n_{L'}) = \begin{cases} 1 & \text{if } n_L \overset{LL}{\leftrightarrow} n_{L'} \text{ or } n_L = n_{L'}, \\ 0 & \text{otherwise} \end{cases} \quad (8.5)$$

where n_L and $n_{L'}$ are considered to be an equivalent entity if they are connected by cross-language links in Wikipedia, denoted by $n_L \overset{LL}{\leftrightarrow} n_{L'}$. Given a key term t , for which the language is not specified, we calculate the matching score of entity $n_{L'}$ based on the maximal probability $P(n_{L'}|t_L)$ as

$$S_m(n_{L'}, t) = \max_{L \in \mathcal{L}} P(n_{L'}|t_L) \quad (8.6)$$

In addition, for each key term t_L in language L that could be a concept name, we first map t_L to the matching concepts C_L in the same language L and then expand each C_L into a set of associated entities in the target language L' , denoted by $N_{L'}^{t_L}$, based on the associations between entities and concepts as well as the cross-language links between entities available in the KB. Let $|N_{L'}^{t_L}|$ denote the number of entities in $N_{L'}^{t_L}$. For each entity $n_{L'} \in N_{L'}^{t_L}$, we calculate its score based on a uniform distribution over all entities in $N_{L'}^{t_L}$. Similarly, the matching score of entity $n_{L'}$ is calculated based on the maximal score w.r.t. t_L as

$$S_m(n_{L'}, t) = \max_{L \in \mathcal{L}} \frac{1}{|N_{L'}^{t_L}|} \quad (8.7)$$

8.3.3. Query Entity Graph Score

Given a key term set T extracted from a keyword query Q and the set of matching entities N_T containing one for each key term $t \in T$, each QEG, denoted by G_Q^T , is constructed from a set of paths that start at each $n_s \in N_T$ matching a key term $t \in T$ and meet at a *connecting element* n_c . Based on that, we introduce a scoring function to assess the relevance of QEGs as follows

$$S(G_Q^T) = \sum_{n_s \in N_T} S(T) \cdot S_m(n_s, t) \cdot S(P_{n_s \rightsquigarrow n_c}) \quad (8.8)$$

where $S(T)$ is the score of key term set T defined in Eq. 8.3, $S_m(n_s, t)$ is the matching score of entity n_s defined in Eq. 8.6 and Eq. 8.7, and $S(P_{n_s \rightsquigarrow n_c})$ captures the score of edges $\langle n_i, n_j \rangle$ along the path $P_{n_s \rightsquigarrow n_c}$ from n_s to n_c , defined as

$$S(P_{n_s \rightsquigarrow n_c}) = \prod_{\langle n_i, n_j \rangle \in P_{n_s \rightsquigarrow n_c}} \frac{S_r(n_i, n_j) \cdot (S_p(n_i) + S_p(n_j))}{2} \quad (8.9)$$

where $S_r(n_i, n_j)$ measures the relatedness between entities n_i and n_j , and $S_p(n)$ reflects the popularity of entity n .

For each pair of entities n_i and n_j , we adopt the Wikipedia link-based measure described in (Milne & Witten, 2008a) to calculate their relatedness score as follows

$$S_r(n_i, n_j) = 1 - \frac{\log(\max(|N_i|, |N_j|)) - \log(|N_i \cap N_j|)}{\log(|N|) - \log(\min(|N_i|, |N_j|))} \quad (8.10)$$

where N_i and N_j are the sets of entities that link to n_i and n_j respectively, and N is the set of all entities in the KB.

To measure entity popularity, we exploit both Wikipedia link structure and page view statistics. The second source captures the number of times Wikipedia pages are requested and can be treated as a query log of entities. By leveraging the two sources, we calculate the frequency of entity n as

$$freq(n) = freq_{link}(n) + \beta \cdot freq_{view}(n) \quad (8.11)$$

where $freq_{link}(n)$ denotes the number of links pointing to n in Wikipedia and $freq_{view}(n)$ denotes the average number of page view requests on n per day. While $freq_{link}(n)$ represents the prior popularity of n in the KB, $freq_{view}(n)$ captures the popularity of n based on user interests. Due to the different scales between Wikipedia link frequency and page view request frequency, $freq_{view}(n)$ is adjusted by a balance parameter $\beta = \frac{\text{total number of links in Wikipedia}}{\text{average number of page views per day}}$, which accounts for the difference in frequencies of Wikipedia links and per-day page view requests. Then the popularity score of each entity $n \in N$ is calculated as

$$S_p(n) = \frac{freq(n)}{\sum_{n_i \in N} freq(n_i)} \quad (8.12)$$

8.4. Top-k Query Graph Exploration

In this section, we present the top- k query graph exploration for *efficient query interpretation generation*. The goal is to find top- k QEGs that connect at least one entity for each key term in a key term set. For pragmatic reasons, existing solutions (He et al., 2007; Li et al., 2008; Tran et al., 2009) use a maximal path length d_{max} , such that only paths of length d_{max} or less between entities n_i and n_j , denoted by $n_i \leftrightarrow^{d_{max}} n_j$, will be taken into account. Such restriction has also been applied to graph exploration in this work, where d_{max} is set as 6. The algorithm is shown in Alg. 2.

Input and Data Structures. The input to the algorithm comprises the list of top- m key term sets $LT = \{T_1, \dots, T_m\}$ and the list $LN = \{N_{t_1}, \dots, N_{t_n}\}$, where each N_{t_i} is a set of entities matching key term t_i . And d_{max} is the maximal path length applied to the graph exploration. For each entity n , we keep track of the information of paths from an entity n_{start}

Algorithm 2: Top-k Exploration of QEGs

Input: $LT = \{T_1, \dots, T_m\}$; $LN = \{N_{t_1}, \dots, N_{t_n}\}$; d_{max} .
Data: $n.S_{t_j^i} = \{\langle n_1, s_{n_1} \rangle, \dots, \langle n_l, s_{n_l} \rangle\}$; $n.s_{t_j^i}$; $n.d_{t_j^i}$; $LQ_{T_i} = \{NQ_{t_1^i}, \dots, NQ_{t_{|T_i|}^i}\}$;
 $UB_{T_i} = \{ub_{t_1^i}, \dots, ub_{t_{|T_i|}^i}\}$; $S(G_Q^{T_i})$; R ; θ .

Result: the top-k optimal QEGs.

```

1  foreach  $T_i \in LT$  do
2      foreach  $t_j^i \in T_i$  do
3          foreach  $n_{start} \in N_{t_j^i}$  do
4              if  $\forall t_{k \neq j}^i \in T_i, \exists n'_{start} \in N_{t_k^i}: n_{start} \overset{d_{max}}{\rightsquigarrow} n'_{start}$  then
5                   $s_{n_{start}} \leftarrow S(T_i) \cdot S_m(n_{start})$ ;
6                   $n_{start}.S_{t_j^i}.add(\langle n_{start}, s_{n_{start}} \rangle)$ ;
7                   $n_{start}.s_{t_j^i} \leftarrow s_{n_{start}}$ ;
8                   $n_{start}.d_{t_j^i} \leftarrow 0$ ;
9                   $NQ_{t_j^i}.add(n_{start})$ ;
10             end
11         end
12          $ub_{t_j^i} \leftarrow \max_{n \in NQ_{t_j^i}} n.s_{t_j^i}$ ;
13     end
14      $\overline{S(G_Q^{T_i})} \leftarrow \sum_{ub_{t_j^i} \in UB_{T_i}} ub_{t_j^i}$ ;
15 end
16 while not all  $NQ \in LQ$  are empty do
17      $T_i \leftarrow \arg \max_{T_i \in LT} \overline{S(G_Q^{T_i})}$ ;
18      $t_j^i \leftarrow \arg \max_{t_j^i \in T_i} ub_{t_j^i}$ ;
19      $n \leftarrow NQ_{t_j^i}.pop()$ ;
20     foreach  $n' \in n.neighbors()$  do
21          $n'.d_{t_j^i} \leftarrow n.d_{t_j^i} + 1$ ;
22         if  $n'.d_{t_j^i} < d_{max}$  and  $\forall t_{k \neq j}^i \in T_i, \exists n'_{start} \in N_{t_k^i}: n' \overset{d_{max} - n'.d_{t_j^i}}{\rightsquigarrow} n'_{start}$  then
23             foreach  $\langle n_{start}, s_{n_{start}} \rangle \in n.S_{t_j^i}$  do
24                  $s'_{n_{start}} \leftarrow s_{n_{start}} \cdot \frac{S_r(n, n') \cdot (S_p(n) + S_p(n'))}{2}$ ;
25                  $n'.S_{t_j^i}.add(\langle n_{start}, s'_{n_{start}} \rangle)$ ;
26             end
27              $n'.s_{t_j^i} \leftarrow n'.S_{t_j^i}.maxScore()$ ;
28              $NQ_{t_j^i}.add(n')$ ;
29              $ub_{t_j^i} \leftarrow \max_{n \in NQ_{t_j^i}} n.s_{t_j^i}$ ;
30              $\overline{S(G_Q^{T_i})} \leftarrow \sum_{ub_{t_j^i} \in UB_{T_i}} ub_{t_j^i}$ ;
31             if  $\forall t_j^i \in T_i: n'.S_{t_j^i}$  is not empty then
32                  $R.add(\text{newQEGsByMergingPath}(n'))$ ;
33                 if  $R.size() \geq k$  and  $\max_{T_i \in LT} \overline{S(G_Q^{T_i})} < \theta$  then
34                     return Top-k( $R$ );
35                 end
36             end
37         end
38     end
39 end
40 return Top-k( $R$ );

```

8. Cross-lingual Keyword Query Interpretation

matching $t_j^i \in T_i^5$ to n , where $n.S_{t_j^i}$ is used to store each pair of the starting entity n_{start} and the score $s_{n_{start}}$ of the path from n_{start} to n , $n.s_{t_j^i}$ and $n.d_{t_j^i}$ are employed to store the maximal score extracted from $n.S_{t_j^i}$ and the length of shortest path from entities matching t_j^i to n , respectively. For each T_i , LQ_{T_i} is a list of $NQ_{t_j^i}$, each of which is a priority queue of entities on the paths starting at entities matching t_j^i and UB_{T_i} is a list of upper bound scores $ub_{t_j^i}$ for paths starting at entities matching all $t_j^i \in T_i$. For supporting top- k , R is used to keep track of the obtained candidate QEGs during graph exploration and θ denotes the lowest top- k score of the QEG in R .

Initialization. Instead of starting at entities matching each query keyword as described in (Kacholia et al., 2005; He et al., 2007; Li et al., 2008; Tran et al., 2009), our exploration starts with each matching entity $n_{start} \in N_{t_j}$ for a key term $t_j^i \in T_i$ (Line 1-3). For each starting entity n_{start} , we first check its connectivity (Line 4) to avoid unproductive exploration, which will be discussed later. When the connectivity condition is satisfied, we initialize the score $s_{n_{start}}$ stored in $n_{start}.S_{t_j^i}$, the maximal score $n_{start}.s_{t_j^i}$ and the distance $n_{start}.d_{t_j^i}$ (Line 5-8). Such starting entities n_{start} are then added into the respective queue $NQ_{t_j^i} \in LQ_{T_i}$ (Line 9) and the upper bound score $ub_{t_j^i}$ for each t_j^i is initialized as the maximal score for all $n_{start} \in NQ_{t_j^i}$ (Line 12).

Connectivity Checking. The aim of checking the connectivity (Line 4 and Line 22) is to predict whether an entity n could participate in any QEGs. Given an entity n with path of length $n.d_{t_j^i}$ from n_{start} matching $t_j^i \in T_i$ to n , if it cannot reach some entities n'_{start} matching $t_k^i \in T_i$ ($k \neq j$) within distance $d_{max} - n.d_{t_j^i}$, it is guaranteed not to be a connecting element and thus the exploration involving n can be avoided. For efficient entity connectivity indexing, we model paths between entities in G_{KB} with length no larger than d as a boolean matrix M_{KB}^d , where each entry m_{ij}^d is 1, if there is a path between entities n_i and n_j of length no larger than d ; otherwise, m_{ij}^d is 0. The matrix $M_{KB}^{d_{max}}$ is constructed iteratively using the formula $M_{KB}^{d_{max}} = M_{KB}^{d_{max}-1} \times M_{KB}^1$.

Upper Bound Principle. The upper bound principle captures the goal of exploring only necessary entities for generating the top- k QEGs. The key is to effectively bound the ultimate score of potential QEGs based on the currently explored paths. Since the score of each edge $\langle n_i, n_j \rangle$ defined in Eq. 8.9 is less than 1, the score of paths satisfy the subset monotonic property, namely $S(P_{n_{start} \rightsquigarrow n}) \geq S(P_{n_{start} \rightsquigarrow n'})$ if $P_{n_{start} \rightsquigarrow n} \subseteq P_{n_{start} \rightsquigarrow n'}$. This implies that the score of a path cannot increase after path expansion during graph exploration and thus the score of all paths starting at entities matching t_j^i can be upper bounded by the maximal score for all $n \in NQ_{t_j^i}$. i.e., $ub_{t_j^i} = \max_{n \in NQ_{t_j^i}} n.s_{t_j^i}$, where $n.s_{t_j^i} = n.S_{t_j^i}.maxScore()$. These upper bound scores indicate the best the potential QEGs resulting from T_i , denoted

⁵We use t_j^i to denote a key term t_j belonging to a specific key term set T_i , while t_j represents the same key term without considering the key term sets it belongs to.

by $G_Q^{T_i}$, can eventually achieve, such that we define the maximal possible score for all $G_Q^{T_i}$ as $\overline{S(G_Q^{T_i})} = \sum_{ub_{t_j^i} \in UB_{T_i}} ub_{t_j^i}$, which will guide our graph exploration and help with early termination.

Graph Exploration. The graph exploration starts with entities in $NQ \in LQ$ (Line 16). To avoid the unnecessary exploration, our algorithm prioritizes the entity by the maximal possible score of the potential QEGs. At each iteration, the most promising T_i that could result in the optimal QEG and the key term $t_j^i \in T_i$ with the largest upper bound score $ub_{t_j^i}$ are selected (Line 17-18). Then the entity n achieving the maximal score of paths from entities matching t_j^i to n is taken from $NQ_{t_j^i}$ (Line 19) and the algorithm continues to explore the neighborhood of n , i.e., all adjacent entities n' . In case that the distance $n'.d_{t_j^i}$ does not exceed d_{max} and the connectivity condition is satisfied (Line 22), we expand the path from each n_{start} to n by adding n' , and the score $s'_{n_{start}}$ of each expanded path is calculated and added into $n'.S_{t_j^i}$ (Line 24-25), where the maximal score $n'.s_{t_j^i}$ is extracted (Line 27). All newly explored entities n' are then added into $NQ_{t_j^i}$ for further exploration (Line 28). Since the maximal score of paths from entities matching t_j^i might change after expansion, the upper bound score $ub_{t_j^i}$ and the maximal possible score $\overline{S(G_Q^{T_i})}$ of potential QEGs are updated accordingly (Line 29-30). If n' is verified to be an connecting element, i.e., for all $t_j^i \in T_i$, there exists a path from n_{start} matching t_j^i to n' (Line 31), the new QEGs generated by merging paths resulted from n' are added into R (Line 32). Finally, we check whether the exploration can terminate to retrieve the top- k QEGs (Line 33-35), which will be discussed in the following.

Early Termination. The exploration terminates when one of the following conditions is satisfied: (1) all possible entities have been explored such that there are no further entities in any $NQ \in LQ$ or (2) the top- k QEGs are guaranteed to be obtained. With the goal of retrieving the top- k QEGs, all entities have to be considered as connecting element in order to keep track of all possible QEGs. However, the upper bound principle deals with the requirement of early termination. The maximal possible score $\overline{S(G_Q^{T_i})}$ for all T_i indicates the best the potential QEGs can achieve and the lowest top- k score of the obtained QEGs captures the threshold θ such that only the QEGs with score higher than or equal to θ have a chance to make into the top- k . To conclude that the current k top-ranked QEGs in R are guaranteed to qualify for the final top- k and thus the exploration can terminate, there should be at least k QEGs in R and $\overline{S(G_Q^{T_i})}$ for all T_i must be below θ , i.e., $\max_{T_i \in LT} \overline{S(G_Q^{T_i})} < \theta$ (Line 33-35).

8.5. Experiments

The experiments were conducted on a virtual machine with 8 Cores at 2.0GHz and 40GB memory and our system is implemented in Java 8. To assess both effectiveness and efficiency

of our approach addressed by Sec. 8.3 and Sec. 8.4 respectively, we asked colleagues and students to provide keyword queries along with the underlying information needs. It results in 21 English queries, 10 German queries, 5 Chinese queries and 14 multilingual queries⁶, where the query length ranges from 2 to 7 with an average of 3.24. We assume that the language of each keyword query is unknown and the target language of query interpretations is English⁷.

8.5.1. Effectiveness Evaluation

For evaluating the effectiveness of query interpretation ranking, which is mainly addressed by Sec. 8.3, we consider the normalized Discounted Cumulative Gain at rank k , denoted by $nDCG@k$, as quality criterion, which measures the goodness of a retrieval model based on the graded relevance of the top- k results. According to our query interpretation problem, the results are judged by the students who provide the keyword queries on 0-5 relevance scale based on the criteria such as relevance, completeness and correctness w.r.t. the underlying information needs.

For a comparative analysis, we conducted the experiments with the following approaches: (1) the baseline using an online *machine translation* service⁸ and a *keyword-based scoring* function described in (Tran et al., 2009), denoted by *MT+KS*; (2) the baseline using our *cross-lingual lexica* for keyword-to-entity mapping and the *keyword-based scoring* same as (1), denoted by *CL+KS*; (3) the baseline using the *machine translation* service same as (1) and an adaption of our query entity *graph scoring* based on *key term* sets, denoted by *MT+GS+KT*; (4) our approach using the *cross-lingual lexica* for entity matching and the query entity *graph scoring* based on *key term* sets as discussed in Sec. 8.3, denoted by *CL+GS+KT*.

Fig. 8.2(a) illustrates the $nDCG@20$ of different approaches for the individual queries (Q1-Q50). Our approach *CL+GS+KT* achieves the best results for 38 queries, while *MT+KS*, *CL+KS* and *MT+GS+KT* perform the best for 9, 16 and 28 queries, respectively. Comparing the two methods with keyword-based scoring function, i.e., *MT+KS* and *CL+KS*, it is observed that using our cross-lingual lexica (*CL*) performs better than the machine translation service (*MT*) in most cases (e.g., Q10-Q14). There is a similar conclusion for the approaches based on our query entity graph scoring, i.e., *MT+GS+KT* and *CL+GS+KT* (e.g., Q27-Q31). Based on the further comparison between *MT+KS* and *MT+GS+KT* as well as *CL+KS* and *CL+GS+KT*, our query entity graph scoring based on key term sets (*GS+KT*) considerably outperforms the keyword-based scoring (*KS*) (e.g., Q38-Q50). By taking advantage of both

⁶It is a realistic phenomenon that queries consist of keywords in different languages, especially for Chinese users, which is also reflected in the 14 multilingual queries in our experiments, where only English and Chinese keywords are contained.

⁷In our experiments, we use English as the target language of query interpretations, but it can be easily extended to other languages.

⁸In our experiments, we used GOOGLE TRANSLATE for translating queries in different languages to English by selecting the input language option as "Detect language".

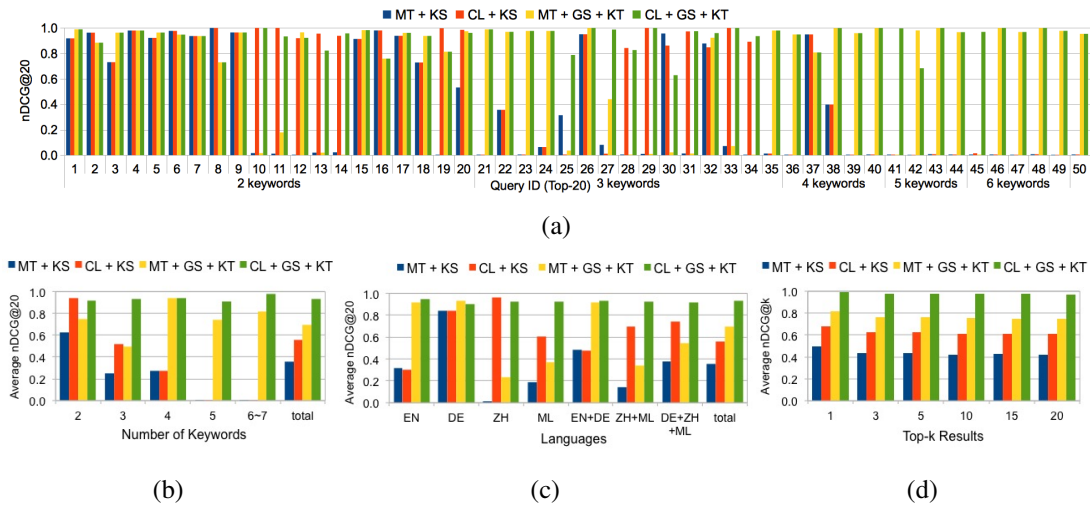


Figure 8.2.: Experimental results of query interpretation effectiveness.

CL and GS+KT compared with MT and KS, CL+GS+KT apparently achieves the best results in most cases.

Fig. 8.2(b) illustrates the impact of query length l , i.e., the number of keywords, on query interpretation effectiveness. While our approach CL+GS+KT is stable for different l , the results of other approaches change considerably when l varies. More specifically, the performance of the approaches using keyword-based scoring (KS), i.e., MT+KS and CL+KS, decreases rapidly when l increases. This is due to the fact that when l is larger, the query entities are usually expressed by more than one keyword such that the keyword-to-entity mapping doesn't work well.

The impact of languages on query interpretation is shown in Fig. 8.2(c). For English queries (EN), by comparing MT+KS with CL+KS and MT+GS+KT with CL+GS+KT, MT and CL exhibit only minor differences because no cross-lingual mapping is needed when the input and target languages are both English. However, MT+GS+KT and CL+GS+KT still considerably outperform MT+KS and CL+KS respectively, because GS+KT has a clear advantage over KS. For German queries (DE), all approaches achieve comparable results for two reasons: (1) the entities in German queries are usually expressed by compound keywords or their abbreviations, e.g., “Fußball-Weltmeisterschaft” or “WM” corresponding to “FIFA World Cup”, such that the keyword-based scoring yields a similar performance to ours; (2) the machine translation service works well when translating from German to English. For Chinese queries (ZH), CL+KS and CL+GS+KT considerably outperform MT+KS and MT+GS+KT because the machine translation service (MT) doesn't work well for translating entity names from Chinese to English compared with our cross-lingual lexica (CL). In addition, in Chinese queries each en-

tity is usually split by users as one compound keyword such that CL+KS even yields slightly better results than CL+GS+KT. Obviously, CL+GS+KT achieves the best results for multilingual queries (ML), where MT+KS and MT+GS+KT perform the worst because the machine translation service (MT) cannot deal with the keywords in multiple languages simultaneously. The experimental results for different combinations of the query languages are also shown in Fig. 8.2(c), where our approach CL+GS+KT achieves the best results (with $nDCG@20 > 0.9$) for most cases.

Fig. 8.2(d) illustrates the results of $nDCG@k$ for different k . We observe that the performance of all approaches decreases slightly when k becomes larger. Among these approaches, CL+GS+KT achieves the most stable performance, e.g., MT+KS, CL+KS, MT+GS+KT and CL+GS+KT yield 15%, 10%, 8% and 2% performance degradation respectively, when k varies from 1 to 20.

8.5.2. Efficiency Study

For assessing the efficiency of query interpretation generation, which is mainly addressed by Sec. 8.4, we conducted the experiments with several approaches: (1) the *keyword-based exploration* from each keyword matching entity (Kacholia et al., 2005), denoted by *KE*; (2) the *top-k algorithm* on top of the *keyword-based exploration* (Tran et al., 2009), denoted by *KE+Top-k*; (3) our key term *set-based exploration* starting from the entities matching the extracted key terms, denoted by *SE*; (4) our graph exploration incorporating only *connectivity checking*, denoted by *SE+CC*; (5) our graph exploration incorporating only *early termination*, denoted by *SE+ET*; (6) our approach incorporating both *connectivity checking* and *early termination* into the graph exploration as discussed in Sec. 8.4, denoted by *SE+CC+ET*.

We start with a comparison between different approaches for the individual queries. The experimental results for computing the top-20 query interpretations for Q21-Q50 with query length from 3 to 7 are illustrated in Fig. 8.3(a). For the sake of space, we omit the results for Q1-Q20 with query length 2, where individual times do not exhibit significant differences. Clearly, SE outperforms KE for the long queries (e.g., Q36-Q50), where 42% performance improvement has been achieved on average, while the performance of SE for short queries is slightly better than KE (e.g., Q21-Q35) or similar to KE (e.g., Q1-Q20). Such differences are primarily due to the number of starting entities for the graph exploration as shown in Fig. 8.3(b). While both connectivity checking (CC) and early termination (ET) contribute to the performance improvement individually, the incorporation of both of them into SE yields the best results. Compared with the baselines KE and KE+Top-k, our approach SE+CC+ET achieves a considerable performance improvement in most cases.

We have investigated the impact of query length l on the performance of different approaches. Fig. 8.3(c) shows the average processing time for different l . Compared with KE, the processing time for SE is relatively stable. The reason might be the number of starting entities

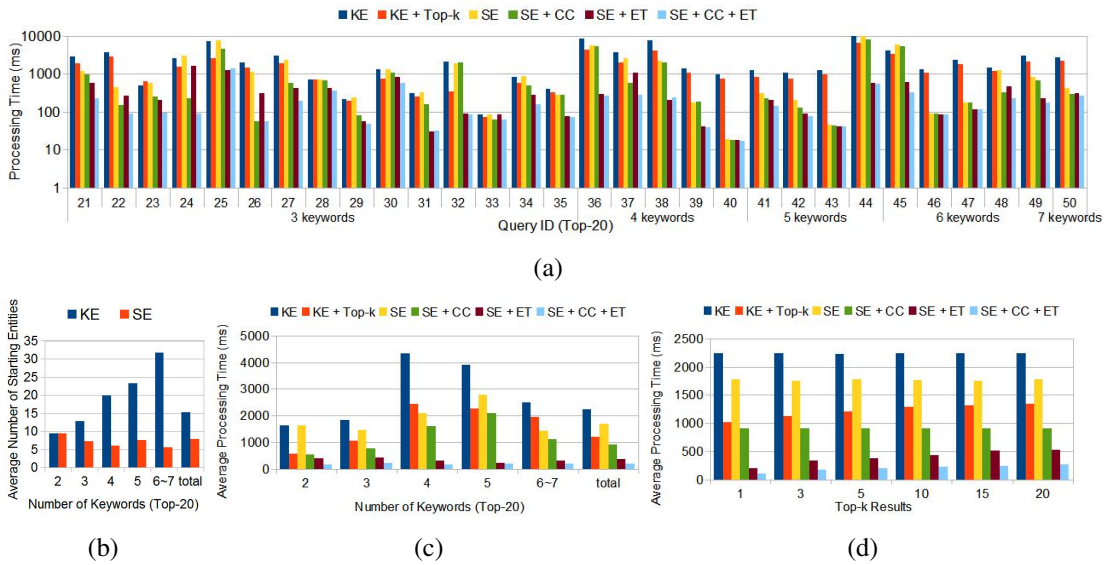


Figure 8.3.: Experimental results of query interpretation efficiency.

generated by SE is less sensitive to l as shown in Fig. 8.3(b). Furthermore, our approaches SE+ET and SE+CC+ET are not sensitive to l due to the application of early termination (ET), while the performance of other approaches changes with varying l .

Fig. 8.3(d) shows the average time for computing top- k query interpretations for different k . The time needed by KE+Top- k , SE+ET and SE+CC+ET decreases rapidly when k becomes smaller. For example, KE+Top- k , SE+ET and SE+CC+ET yield 24%, 61% and 62% time reduction respectively, when k varies from 20 to 1, while the performance of other approaches doesn't change with k since they have to process all results no matter what the value of k is. In total, our approach SE+CC+ET outperforms KE by one order of magnitude and is 5 times faster than KE+Top- k when $k = 20$, and it achieves even more considerable performance improvement for smaller k , e.g., 22 times and 10 times faster than KE and KE+Top- k respectively, when $k = 1$.

8.6. Related Work

We firstly present the related work to keyword query interpretation and then review some existing work on cross-lingual and concept-based IR.

Keyword Query Interpretation. The main challenges in dealing with keyword queries are *ambiguity* and *incompleteness* of keywords. The use of thesauri to deal with the ambiguity of keywords has a long history. Most commonly, WordNet thesaurus has been found beneficial

in disambiguating keywords and in choosing their senses (Voorhees, 1994). There are also proposals for mapping keyword queries to elements in an ontology (Tran et al., 2007), where the resulting semantics provides the basis for identifying users' search intents. In addition, graph-based approaches (Kacholia et al., 2005; He et al., 2007; Tran et al., 2009) have been widely used to find substructures in structured data, including relational, XML and RDF data. The recent work (Demidova et al., 2012a; Demidova et al., 2013) also aimed to boost the scalability of interactive query construction over large scale data from the perspective of both user interaction cost and performance.

While existing methods mostly deal with individual keywords in the query, our approach relies on the extracted key terms referring to entities in KBs, which helps to improve both efficiency and effectiveness as shown in our experiments. In addition, the *cross-linguality* issue has not been studied in the previous work.

Cross-lingual and Concept-based IR. Traditional IR is normally based on the bag-of-words (BOW) models, which have the limitation of retrieving only the syntactically relevant but not the semantically relevant documents. Meanwhile, they suffer from the vocabulary mismatch problem, i.e., queries and documents, which are semantically very related, might contain only few terms in common. This problem is more serious in cross-lingual IR due to the fact that queries and documents in different languages rarely share common terms. In order to address the problem, different concept-based solutions (Wei & Croft, 2006; Bendersky & Croft, 2008; Gabrilovich & Markovitch, 2007; Egozi et al., 2011) and their cross-lingual extensions (Sorg & Cimiano, 2008; Potthast et al., 2008) have been proposed. Instead of the BOW models used in the classic IR, the goal is to capture queries and documents as concepts, such that the relevance can be estimated in the concept space even in the presence of vocabulary gap, especially for cross-lingual IR.

Unlike the previous studies, our approach to cross-lingual query interpretation enables entity-based cross-lingual IR by exploiting multilingual knowledge bases, where users can issue keyword queries in any language (even in multiple languages), for retrieving documents related to the query entities in any other languages.

8.7. Conclusions

We present a knowledge base approach to cross-lingual query interpretation by transforming keywords in different languages to their semantic representation. As the main contributions of this work, we propose a scoring mechanism for effective query interpretation ranking and a top- k graph exploration algorithm for efficient query interpretation generation. A separate evaluation on each of these two aspects has been performed and it shows that our approach achieves promising results w.r.t. both effectiveness and efficiency. As future work, we would like to extend our approach by taking into account entity relations expressed in keyword queries to construct the QEGs. And it is essential to perform further evaluation to

show the promising results of our query interpretation can carry over to the performance of cross-lingual IR.

8. *Cross-lingual Keyword Query Interpretation*

9. A Framework of Cross-lingual Semantic Annotation and Search

Modern Web search engines still have many limitations: search terms are not disambiguated, search terms in one query cannot be in different languages, the retrieved Web contents have to be in the same language as the search terms and results are not integrated across a live stream of different media channels, including online news, social media and Live-TV. The framework described in this chapter enables all of this by combining a media stream processing architecture with cross-lingual semantic annotation and search.

9.1. Introduction

The amount of entities in large knowledge bases (KBs) has been increasing rapidly, enabling new ways of semantic information access, like keyword and semantic queries over entities and concepts mentioned in heterogeneous media items. While *entity search* has become a standard feature, Web search engines are still limited in their semantic processing capabilities: it is not possible to disambiguate search terms according to users' intent, search terms in one query cannot be in different languages, the retrieved media items have to be in the same language as the search terms and are not collected across a live stream of different media channels.

This chapter provides an overview of the main components that we developed in the xLiMe project¹ for the cross-lingual processing of media content using *explicit cross-lingual semantics* (Zhang et al., 2017). In particular, we present a framework that intends to break the barriers in between languages and modalities for a seamless semantic access to media streams. Firstly, we present *X-LiSA* (Zhang & Rettinger, 2014), an infrastructure for *multilingual and cross-lingual semantic annotation*, which supports interfaces for annotating multilingual text extracted from different media channels with resources from KBs. It helps to bridge the ambiguity of natural language text and its formal semantics as well as to transform documents in different languages into a unified representation. Based on *X-LiSA*, we then present *XKnowSearch!* (Zhang et al., 2016a), a novel system for entity-based *multilingual and cross-lingual semantic search* by translating keyword queries in different languages to their semantic representation. It bridges the language barriers between queries and documents in different languages, and also facilitates query disambiguation and expansion.

¹<http://www.xlime.eu>

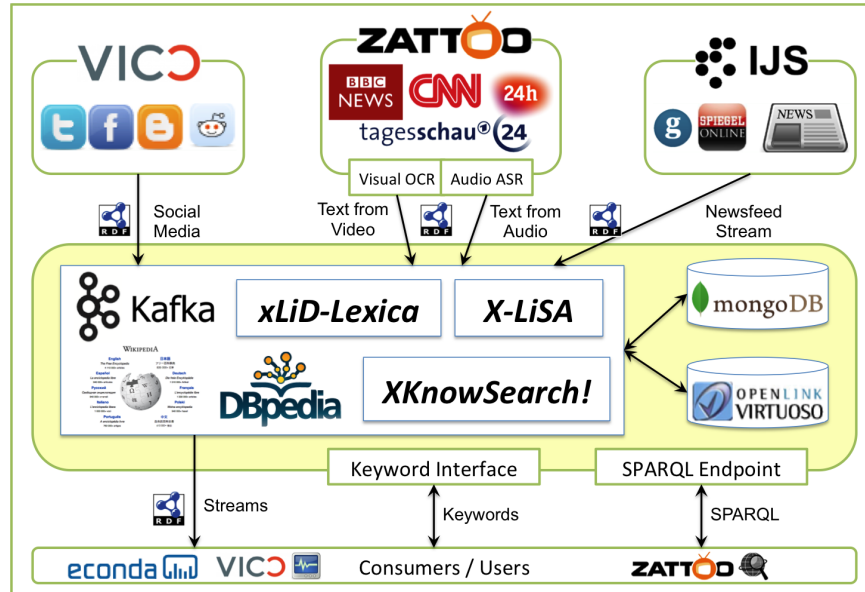


Figure 9.1.: The xLiMe architecture.

The rest of this chapter is structured as follows. We first introduce a real-time media processing architecture and an annotation data model in Sec. 9.2. Then, we present the component for cross-lingual semantic annotation of multilingual text extracted from multiple channels in Sec. 9.3. The annotated cross-channel media stream allows cross-lingual and cross-modal semantic search, which will be described in Sec. 9.4. Finally, we discuss the related work in Sec. 9.5 before we conclude in Sec. 9.6

9.2. Cross-lingual and Cross-modal Processing of Media Streams

In this section, we introduce the architecture of cross-lingual and cross-modal media stream processing as well as the annotation data model developed in the xLiMe project (Zhang et al., 2017). The processing of different multimedia streams is a cost-intensive task, which has been best performed in a distributed manner. Unfortunately, the various sources, their individual particularities, and their distributed processing pose a huge challenge for data integration. As such, we consider three main issues for cross-lingual and cross-modal processing of media streams: (1) multimedia sources; (2) intelligent processing and (3) semantic integration.

The *multimedia sources* include online news, social media and live-TV. All of these sources are multilingual media streams with different and—in the case of social media—changing

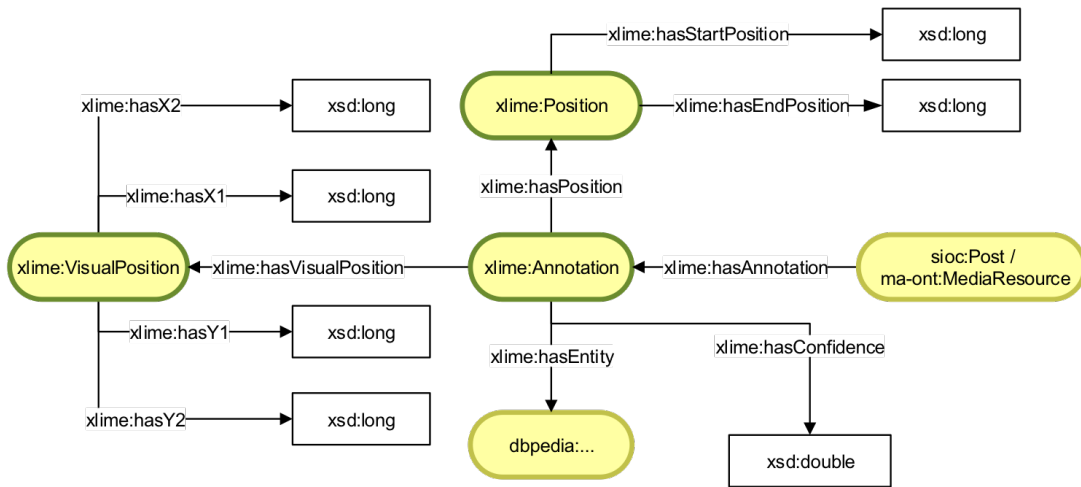


Figure 9.2.: The xLiMe annotation data model.

velocity. The *intelligent processors* include several tools for cross-lingual and cross-modal processing of these sources, in particular *X-LiSA*, our cross-lingual semantic annotation tool for text, which will be discussed in Sec. 9.3. Regarding the live-TV, some preprocessors have been used for extracting text from video (i.e., optical character recognition), accompanied by speech-to-text processing (i.e., automatic speech recognition) in the case of audio streams of TV content. The *semantic integration* of different media streams poses the challenge to identify a common model that suits the diversity of the data sources and the output of the processing engines. Further, we combine the processed data with additional background knowledge to help with cross-lingual and cross-modal semantic search in our *XKnowSearch!* system, which will be discussed in Sec. 9.4.

The architecture of the xLiMe project is divided into multiple components (see Figure 9.1). For practical reasons, the multimedia source and initial processing infrastructure is respectively attached to the partners that provide the respective data. Raw data as well as (intermediately or fully) processed data is directly sent to an ApacheTM Kafka message broker that enables a multitude of different communication channels. The partners that have data processing capabilities provide meta and provenance data in accordance to the xLiMe data model (that will be introduced later). As also the raw data is pushed to the message broker, every partner that has processing capabilities can provide enhanced or alternative services. Along with the message broker, a triple store (i.e., Virtuoso) and a NoSQL database (i.e., MongoDB) provide further data integration and query capabilities. Like this, individual hooks are subscribed to specific Kafka channels and constantly load data into the respective store.

The xLiMe data model is defined as an RDF vocabulary and tailored specifically to the dif-

ferent modalities: text, audio, and video. It extends other vocabularies such as Dublin Core², SIOC³, and KDO⁴. Its main scheme is depicted in Figure 9.2. Similar to the Web Annotation Model⁵, it enables to relate text and (parts of) video or audio streams to real world entities. The predicates that define the start and stop positions, i.e., *xlime:hasStartPosition* and *xlime:hasEndPosition*, can be used in a flexible manner and may define character positions, in the case of text, or milliseconds/frame numbers in case of audio/video. In order to describe rectangular fragments of videos or images, there is a specific class that defines the visual position, i.e., *xlime:VisualPosition*. In any case, the annotation associated with a media resource through the predicate *xlime:hasAnnotation* should relate to an entity in the knowledge base, such as DBpedia, through the predicate *xlime:hasEntity*. Another particularity of the xLiMe annotation data model—that is not depicted in Figure 9.2—is extensive use of named graphs where we use the W3C provenance data model⁶ in order to provide meta data for the respective processing of one or more media items.

Based on the above data model, the xLiMe triple store is individually queryable and enables restrictions and aggregates on multiple modalities, languages and sources. The same accounts for the NoSQL database. This enables the flexible operation of services that build on live streaming data in combination with additional background knowledge. Based on the integrated data in the triple store and NoSQL database, the xLiMe components enable us to ask complex questions, such as “Which cars produced by Mercedes-Benz were mentioned most in the last two weeks?”, using SPARQL queries and to search for different multilingual media channels using keyword queries in any languages, which will be discussed in the next sections.

9.3. Cross-lingual Semantic Annotation

In this section, we present *X-LiSA* (Zhang & Rettinger, 2014; Zhang et al., 2017), an infrastructure for *cross-lingual semantic annotation*, which supports interfaces for annotating multilingual text extracted from different media channels with entities in knowledge bases. It helps to bridge the ambiguity of unstructured data and its formal semantics as well as to transform such data in different languages into a unified representation.

9.3.1. System Architecture

The architecture of *X-LiSA* is illustrated in Figure 9.3, where *cross-lingual groundings extraction* is performed offline to generate the indexes that are needed by the online *cross-lingual*

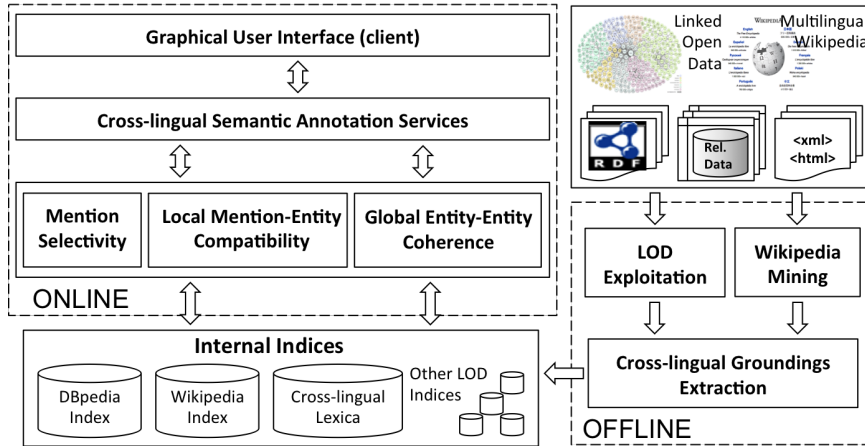
²<http://dublincore.org/documents/dces/>

³<https://www.w3.org/Submission/sioc-spec/>

⁴<http://render-project.eu/resources/kdo/>

⁵<https://www.w3.org/TR/annotation-model/>

⁶<https://www.w3.org/TR/prov-dm/>

Figure 9.3.: The system architecture of *X-LiSA*.

semantic annotation.

Cross-lingual Groundings Extraction. For matching words and phrases in different languages against entities in knowledge bases, *X-LiSA* utilizes *xLiD-Lexica* (Zhang et al., 2014a; Zhang et al., 2014b), our recently established cross-lingual linked data lexica by exploiting various kinds of structures in Wikipedia, such as anchor texts of hyperlinks and cross-language links, to extract the cross-lingual groundings of entities. More details about how to construct *xLiD-Lexica* can be found in Chapter 7.

Mention Detection. The first challenge of semantic annotation lies in *mention selectivity* with the goal of detecting the boundaries of mentions in text that are likely to denote entities. To address the challenges of correctness, completeness and emergence of the detected mentions, we employ our recent work (Zhang et al., 2015a) that aims to detect both named and nominal entities. Such entity mentions serve as the input of entity disambiguation.

Entity Disambiguation. For each mention, its candidate entities are then extracted using *xLiD-Lexica*. While the feature of *mention-entity compatibility* captures the most likely entity behind the mention based on the cross-lingual groundings and the entity that best fits the context of the mention based on the cross-lingual relatedness (Zhang et al., 2015c), *entity-entity coherence* collectively captures the related entities as annotations. These features are then employed by the *graph-based disambiguation* to determine the referent entity for each mention. More details about the disambiguation algorithm can be found in Chapter 3.

9.3.2. Functionality Description

In the following, we discuss *X-LiSA* in terms of the *online annotation service* and the use case of *media annotation and querying*.

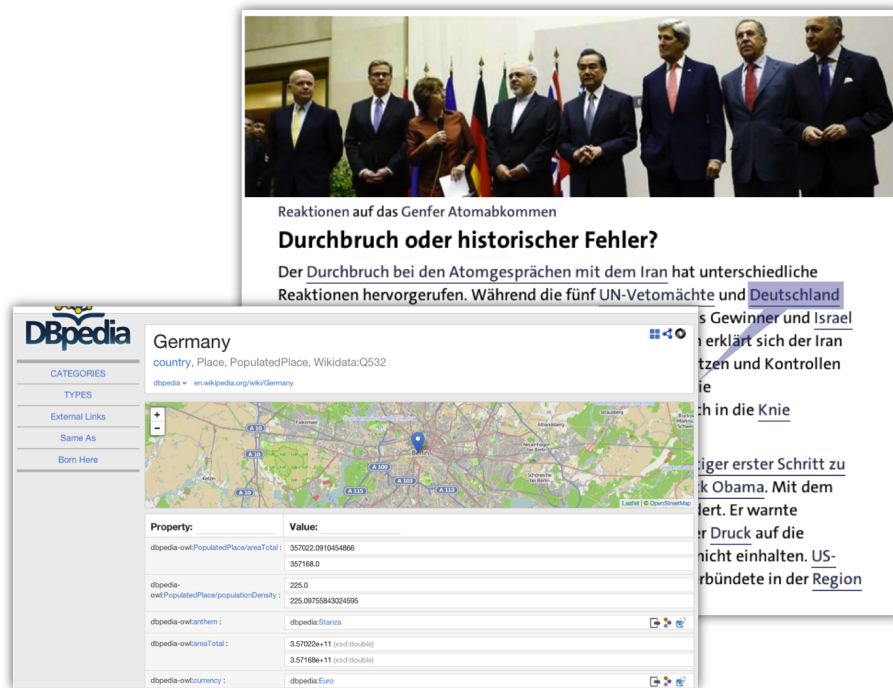


Figure 9.4.: Example of annotation service for Web pages.

Online Annotation Service. *X-LiSA* supports interfaces for annotating raw text and Web pages in different languages. A screenshot of the cross-lingual semantic annotation service is shown Figure 9.4, where the input is the URL of a German news article, the knowledge base is DBpedia and the output language is English. In order to allow not only users but also software agents to access the functionality of text annotation, we also provide the service, which takes raw text as input and yields the output of annotations in XML as shown in Figure 9.5.

Media Annotation and Querying. Within the context of xLiMe project, *X-LiSA* has been widely used to annotate textual data from mainstream news sites, social media and Live-TV, where the following partners have contributed large datasets, which are delivered as streams:

- *JSI NewsFeed*⁷: news articles crawled from online news sites across the world.
- *VICO*⁸: social media text crawled from forums, blogs, social networks, review sites and others.
- *Zattoo*⁹: text extracted from visual and audible TV data based on the technologies, such

⁷<http://newsfeed.ijs.si>

⁸<http://www.vico-research.com/en>

⁹<http://developer.zattoo.com>

```

<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Source>Während die fünf UN-Vetomächte und Deutschland von einem wichtigen Erfolg
sprechen, sehen sich der Iran als Gewinner und Israel als Verlierer der Verhandlungen in
Genf.</Source>
  <Result>Während die fünf [[United_Nations_Security_Council_veto_power|UN-Vetomächte]]
und [[Germany|Deutschland]] von einem wichtigen Erfolg sprechen, sehen sich der Iran als
Gewinner und [[Israel]] als Verlierer der Verhandlungen in [[GenevalGenf]].</Result>
  <Annotations>
    <Annotation URI="http://dbpedia.org/resource/Iran" lang="en" mention="Iran" />
    <Annotation
URI="http://dbpedia.org/resource/United_Nations_Security_Council_veto_power" lang="en"
mention="UN-Vetomächte" />
    <Annotation URI="http://dbpedia.org/resource/Germany" lang="en"
mention="Deutschland" />
    <Annotation URI="http://dbpedia.org/resource/Israel" lang="en" mention="Israel" />
    <Annotation URI="http://dbpedia.org/resource/Geneva" lang="en" mention="Genf" />
  </Annotations>
</Response>

```

Figure 9.5.: Example of annotation service output in XML.

```

<http://vico-research.com/social/30997f51-f09f-353f-9e29-ec04a421fb16>
  a                sioc:MicroPost ;
  dcterms:created  "2014-12-07T11:32:51"^^xsd:dateTime ;
  dcterms:language "en" ;
  dcterms:publisher <http://www.twitter.com/> ;
  dcterms:source   <http://twitter.com/WexMart161/statuses/541540837994561536> ;
  sioc:content     "FOR SALE: A Splendid 1959 #Mercedes 190SL #Convertible in BRAND NEW
condition http://t.co/mSBff9WzVl http://t.co/WVKHdGWjTh" ;
  sioc:has_creator <http://twitter.com/WexMart161> ;

  xlime:hasAnnotation [ xlime:hasConfidence "1"^^xsd:double ;
                        xlime:hasEntity      dbpedia:Mercedes-Benz_190SL ;
                        xlime:hasPosition     [ xlime:hasStartPosition "36"^^xsd:long ;
                                              xlime:hasStopPosition  "41"^^xsd:long ]
                      ] ; ...

```

Figure 9.6.: Example of annotated social media text in RDF (Turtle format).

as optical character recognition (OCR) and automatic speech recognition (ASR).

Based on the xLiMe annotation data model introduced in Sec. 9.2, we model the annotated media data as RDF triples, which are stored in our triple store. An example of annotated social media text is shown in Figure 9.6. In addition, a SPARQL endpoint is provided for querying the annotated data. For example, given the question “Which cars produced by Mercedes-Benz were mentioned most in the last two weeks?”, the SPARQL query shown in Figure 9.7 can be used to answer the question, where the top-10 results are illustrated in Figure 9.8.

9. A Framework of Cross-lingual Semantic Annotation and Search

```
PREFIX xlime: <http://xlime-project.org/vocab/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT COUNT(DISTINCT ?media) as ?count ?entity WHERE {
  ?s xlime:hasAnnotation ?a .
  ?s dcterms:source ?media .
  ?s dcterms:created ?date .
  ?a xlime:hasEntity ?entity .
  ?entity dbpedia-owl:manufacturer dbpedia:Mercedes-Benz .
  FILTER (?date > xsd:date(now()-3600*24*14) && ?date < now()) .
} GROUP BY ?entity ORDER BY DESC(?count)
```

Figure 9.7.: Example of SPARQL query.

count	entity
28120	http://dbpedia.org/resource/Mercedes-Benz_190SL
12329	http://dbpedia.org/resource/Mercedes-Benz_S-Class
7170	http://dbpedia.org/resource/Mercedes-Benz_W123
6363	http://dbpedia.org/resource/Mercedes-Benz_E-Class
4042	http://dbpedia.org/resource/Mercedes-Benz_G-Class
3897	http://dbpedia.org/resource/Mercedes-Benz_SL-Class
3097	http://dbpedia.org/resource/Mercedes-Benz_SLS_AMG
2985	http://dbpedia.org/resource/Mercedes-Benz_M-Class
1966	http://dbpedia.org/resource/Mercedes-Benz_CLK-Class
1495	http://dbpedia.org/resource/Mercedes-Benz_SLK-Class

Figure 9.8.: Example of SPARQL query results.

9.4. Cross-lingual and Cross-modal Semantic Search

X-LiSA offers opportunities for dealing with complex information needs on the media data using the formal query language, i.e., SPARQL,. However, such queries hinder casual users in expressing their information needs as they might be not familiar with the query’s syntax or the underlying data model. Because keyword search are easier to handle for casual users, we present *XKnowSearch!* (Zhang et al., 2016a; Zhang et al., 2017), a novel system for entity-based *multilingual* and *cross-lingual semantic search* by translating keyword queries in different languages to their semantic representation. It bridges the language barriers between keyword queries and media data as well as facilitates query disambiguation and expansion in both manual and automatic manners.

9.4.1. System Architecture

By employing *X-LiSA* for offline semantic annotation, *XKnowSearch!* enables keyword search by capturing keyword queries and media data items at the semantic level and also bridging the language barriers. The architecture of *XKnowSearch!* is shown in Figure 9.9. In the following,

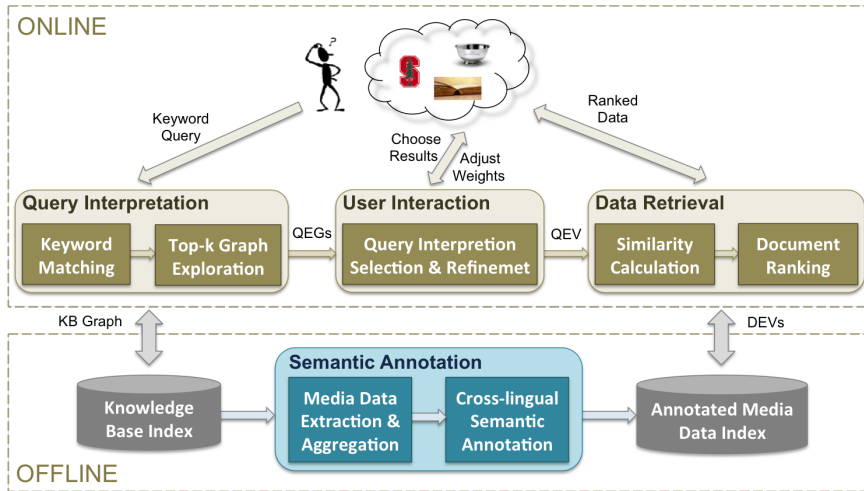


Figure 9.9.: The system architecture of *XKnowSearch!*.

we discuss the online processing components.

Query Interpretation. The search process starts with a keyword query in any language (even with keywords in multiple languages). Instead of retrieving media items directly by keywords, *XKnowSearch!* first finds the *query entity graphs (QEGs)* matching the keyword query by exploring the semantic graph of the knowledge base with nodes representing entities and edges describing their relations.

The first step of *query interpretation* is *keyword matching*. To address the challenge of matching query keywords in different languages to entities, we also make use of *xLiD-Lexica* described in Chapter 7. After obtaining the matching entities, the *top-k graph exploration* is then performed on the graph of the knowledge base for finding the top-*k* optimal QEGs. The resulting QEGs represent different semantic interpretations of the keyword query. Thus it can help users to refine the query and influence media item ranking according to the search intents. More details about our approach to query interpretation can be found in Chapter 8.

User Interaction. Different interpretations of the keyword query, i.e., the generated QEGs, are then presented to users for *selecting* the one that fulfills their search intents. The selected QEG can be further *refined*. From an entity in the QEG, users can navigate its description and the connected entities through their relations in the knowledge base, such that they can add additional entities into the QEG or delete unnecessary ones. After that, the entities in the refined QEG constitute the *query entity vector (QEV)*, where each entry contains the weight of the corresponding entity, which is calculated by the top-*k* graph exploration algorithm for query interpretation and can also be adjusted by users. These weights will be leveraged for ranking of retrieved media items in the next component.

We consider *user interaction* as beneficial because it enables the interactive query disambigua-

tion and expansion according to users' search intents. Although refinement can be made more precisely on QEGs than on keywords, user interaction is optional in our system. Users can also search the media items directly without interactive query refinement. In this case, the QEG with highest score obtained by the query interpretation component is selected.

Data Retrieval. For media data retrieval, the entities in the QEV are used to find relevant media items. However, the media items without mentioning the entities in the QEV could also be relevant when they contain entities that are related to the ones in the QEV. Therefore, integrating the related entities into the query can help to cover more complementary information and thus improve the performance of data retrieval. Based on the above observation, we first construct the *expanded query entity vector (EQEV)* by *automatically expanding* the QEV with additional related entities.

For each media data item, we construct the *data entity vector (DEV)*, where the entries contain the confidence scores of the annotations (i.e., the linked entities), which are generated by offline semantic annotation. It is noted that all the entities in both EQEV and DEV are grounded in the same hub language such that they serve as the bridge to overcome the language barrier between keyword queries and media data. The semantic similarity between the EQEV and each DEV can be calculated based on standard similarity measures, such as cosine similarity, which is then used for ranking of retrieved media items.

9.4.2. Functionality Description

In this section, we show four major features of *XKnowSearch!* with the goal of addressing the challenges that traditional keyword search suffers from.

Query Flexibility. While traditional keyword search systems do not allow users to be involved in the search process to perform query refinement, *XKnowSearch!* supports two search modes, namely *direct search* and *indirect search*. The direct search mode performs similar to the current Web search engines like Google. It takes a keyword query as input and retrieves the relevant media items directly without user involvement in the search process. The indirect search mode provides the opportunity for users to understand the meaning of the query entities and the underlying semantic relations between them, such that users are able to refine and extend the information needs. While the direct search enables users to search in a familiar and convenient manner, the indirect search provides users a *more flexible way* to influence the search process according to their search intents.

Query Disambiguation. Keywords are naturally ambiguous and this problem is more serious in the multilingual setting because the same keywords could have different meanings in different contexts or languages. In *XKnowSearch!*, query disambiguation can be performed both automatically and manually. On the one hand, the query interpretation component *automatically eliminates the ambiguity* of the keyword query by taking advantage of the context, i.e., other query entities, and exploiting the semantic graph of the KB to generate the top-*k* QEGs.

XKnowSearch!

Found 100 articles in 3.25 seconds totally.

- Comment on Lord William Wallace writes... How you can make sure we win this referendum by Paul Murray
- Cameron anuncia tras la victoria del 'Brexit' que se ir? en octubre
- Brexit won't hit global growth, but it does make one big difference
- Boris Johnson: The clown who could be king?
- David Cameron's successor as Prime Minister will be chosen by September 2 amid calls for quicker action on Brexit
- Boris Johnson: The clown who could be king?
- The untold story about NHS and Brexit
- Comment on Lord William Wallace writes... How you can make sure we win this referendum by Peter Parsons
- El Reino Unido, una sociedad fragmentada y sumida en la crisis
- Boris Johnson: The clown who could be king?
- 10 Things to Know for Wednesday
- Google: el temor al Brexit rompe el contador de las b?squedas
- Brexit campaigner Boris Johnson says won't run for PM
- Inmigraci?n y econom?a, lo m?s pol?mico en la campa?a del refer?ndum de la UE
- Comment on Lord William Wallace writes... How you can make sure we win this referendum by theakes
- Brexit campaigner Johnson says won't run for PM
- Futbolistas y exjugadores dan su opini?n acerca del "Brexit"
- Comment on Lord William Wallace writes... How you can make sure we win this referendum by Katerina Porter
- Alta participaci?n marca jornada hist?rica en la que Reino Unido decide su futuro en la UE
- Las apuestas brit?nicas sobre el referendo se decantan por la Uni?n Europea
- Michael McAul: How Brexit is a win for Putin
- A cuatro d?as del refer?ndum, el No al 'Brexit' encabeza los sondeos
- El debate pol?tico que marca? el divorcio entre el Reino Unido y la Uni?n Europea
- Futuro incierto para un Reino que ha quedado muy dividido
- Brit?nicos deciden hoy en plebiscito si abandonan la UE
- El ministro de Empleo brit?nico se presenta como candidato a primer ministro
- Comment on Lord William Wallace writes... How you can make sure we win this referendum by Stevan Rose
- Utilizan el atentado de Orlando para justificar el Brexit
- El giro en las encuestas pone al rojo vivo las ?ltimas horas de la campa?a del 'Brexit'

Brexit won't hit global growth, but it does make one big difference
(http://moneyweek.com/brexit-global-growth-and-central-bank-policy/)

Language: en Longitude: Latfuda: Country:
 Retrieved Date: Thu Jun 30 11:28:09 CEST 2016

So what's on the Brexit agenda today?

The FTSE 100 has rebounded to where it was. You'd expect that - it's an international index. Broadly speaking, a falling currency[**Pound sterling**] means a rising stockmarket. Japan is the most glaringly obvious example of this, but it works elsewhere too.

Other markets - the FTSE 250, eurozone equities - are still off their pre-Brexit highs, and the mid-cap index is a bit lower this morning after a rally yesterday.

The pound[**Pound sterling**] is still creeping higher, even although (or perhaps because) everyone expects it to keep falling.

Oh yes, and both of our main political parties are choosing new leaders...

The effect of political turmoil on the UK

Westminster is in a froth. We've got the excitement of a leadership campaign gripping the Tories. Theresa May, Michael Gove and Boris Johnson[**Boris Johnson**] are squaring up, while Liam Fox and Stephen Crabb are also in there.

Meanwhile, Labour leader (still) Jeremy Corbyn is lining up his post- Westminster[**House of Commons of the United Kingdom**] career in celebrity brand endorsements (superglue ads - it's not Saatchi & Saatchi, but I'm thinking: "Nothing clings tighter than Jezza").

All this excitement is of course translating into our papers, who reckon it's the end of days.

As an investor though, you're better off ignoring this stuff. The market certainly is.

Let's take a quick step back here.

Whoever is next to run the Conservative party (and therefore the country), their political philosophy will be business-friendly and desiring of free trade. Immigration will remain a sticky issue, but the reality is that while it might be a dealbreaker for some voters, it's not the priority of any of these potential leaders.

So whoever is in charge will be business-friendly, and most likely a pragmatist. So despite the froth, none of this is an especially big deal for markets.

Figure 9.10.: Example of retrieved news articles for query “英国 boris johson”.

On the other hand, users can also *disambiguate the query manually* by selecting the most appropriate QEG and further refining it. As query interpretation, QEG is more informative and expressive than keywords such that users can obtain information about not only entities but also relations between them.

Query Expansion. The query keywords are often incomplete in the sense that instead of the full entity name, only the aliases, acronyms and misspellings are usually given by users. *XKnowSearch!* supports query keywords matching entities in their *incomplete* forms. In addition, keyword queries might contain concept names representing a set of associated entities. In *XKnowSearch!*, the matching concepts are automatically expanded into sets of individual entities, which has been discussed in Chapter 8 in detail. As query interpretation, QEG is more informative and expressive than keywords such that it can help users to *manually expand* the query by navigating the knowledge graph and adding more intended entities that are used for media data retrieval. The resulting query entities can be *automatically expanded* with additional related entities in *XKnowSearch!*, which are then used for document retrieval. In the retrieved media items, the entities specified manually by users are distinguished with the ones automatically expanded using different colors (cf. Figure 9.10).

Cross-lingual and Cross-modal Search. Modern Web search engines are still limited in their semantic processing capabilities: search terms in one query cannot be in different languages, the retrieved Web contents have to be in the same language as the search terms and results are not integrated across a live stream of different media channels including online news, social media and Live-TV. In this regard, *XKnowSearch!* aims to break the barriers in between

9. A Framework of Cross-lingual Semantic Annotation and Search


XKnowSearch!

Found 91 social medias in 3.60 seconds totally.

- [Brexit leader Boris Johnson rules out bid to ...](#)
- [Boris Johnson's dad Stanley told LBC he is ...](#)
- [Who's the next UK PM? Our poll puts ...](#)
- [EXCL: Boris and Gove promise to scrap VAT ...](#)
- [Brexit would bring Brits, Aussies closer: Johnson https://t.co/pe8YQcLcr Umm! Don't ...](#)
- [Boris Johnson urges Brits to vote Brexit to ...](#)
- [@vote_leave Juncker aide orders Britons not to ...](#)
- [Boris Johnson states that the margin by which ...](#)
- [I'll say this again. In the UK democracy in ...](#)
- [Why a Brexit is good for Africa: Boris Johnson PM ...](#)
- [Boris Johnson urges Brits to vote Brexit to "take back ...](#)
- [Not really a lot of evidence that UK ...](#)
- [Boris Sets Out Vision For Britain After Brexit https://t.co/MA47IOAON8](#)
- [Boris Johnson says he does not believe those ...](#)
- [?? 7UK votes for Brexit: Cameron resigns, markets fall, protestors ...](#)
- [Thank God Boris Johnson pulled out of ...](#)
- [Blair Sees U.K. Rejecting Brexit as He Attacks Boris Johnson ...](#)
- [Someone uploaded Boris Johnson's Brexit speech to Pornhub with an ...](#)
- [Jungle residents claim Brexit will help them sneak into the ...](#)
- [Boris Johnson, the likely next PM of the ...](#)
- [Watch: David Cameron resigns as Prime Minister after ...](#)
- [Boris Johnson says he's happy for Turkey to ...](#)
- [TheGreatDebate: Boris leave-EU closing speech gets standing ovation \(21Jun16\) ...](#)
- [UK gets recession Boris becomes PM Labour changes leader: General Election ...](#)
- [UK gets recession Boris becomes PM Labour changes leader: General Election ...](#)
- [Just had a vision of a world where ...](#)
- [Boris says Brexit for politics despite economic interest but ...](#)
- [Not really a lot of evidence that UK ...](#)
- [Blair Sees U.K. Rejecting Brexit as He Attacks Boris Johns ...](#)
- [Not really a lot of evidence that UK ...](#)
- [Boris Johnson pushes the UK in to the shit, but ...](#)
- [RGP Brexit: Labour party crisis deepens; Boris Johnson calls ...](#)
- [No one wants to be the PM that goes down ...](#)
- [EU referendum: Boris Johnson says Brexit will he ...](#)

http://twitter.com/neupane_ani/statuses/7484836661170135041

Language: en Publisher: Twitter Retrieved Date: Thu Jun 30 01:49:46 CEST 2016



RT @HuffingtonPost: Brexit leader Boris Johnson[Boris Johnson] rules out bid to be Britain[United Kingdom]'s next prime minister https://t.co/b4gJ9S6u9s https://t.co/q0mu7

Figure 9.11.: Example of retrieved social media posts for query “英国 *boris johnson*”.

XKnowSearch!


Found 100 shows in 7.70 seconds totally.

- [CNN INTERNATIONAL - 11553](#)
- [ITV 1 LONDON - 11738](#)
- [SKYNEWS INTL - 11411](#)
- [SKYNEWS INTL - 11332](#)
- [SKYNEWS INTL - 11410](#)
- [SKYNEWS INTL - 11119](#)
- [SKYNEWS INTL - 10956](#)
- [SKYNEWS INTL - 11307](#)
- [SKYNEWS INTL - 11721](#)
- [BBC WORLD SERVICE - 11404](#)
- [ITV 1 LONDON - 11509](#)
- [ITV 1 LONDON - 11320](#)
- [CNN INTERNATIONAL - 11763](#)
- [CNN INTERNATIONAL - 11331](#)
- [SKYNEWS INTL - 11301](#)
- [BBC PARLIAMENT - 10961](#)
- [CNN INTERNATIONAL - 11342](#)
- [SKYNEWS INTL - 10870](#)
- [SKYNEWS INTL - 11828](#)
- [BBC PARLIAMENT - 11445](#)
- [SKYNEWS INTL - 11832](#)
- [BBC ONE - 11867](#)
- [ITV 1 LONDON - 11850](#)
- [SKYNEWS INTL - 11412](#)
- [ITV 1 LONDON - 11737](#)
- [BBC PARLIAMENT - 11031](#)
- [SKYNEWS INTL - 11554](#)
- [BLOOMBERG EUROPE - 11379](#)
- [ITV 1 LONDON - 11185](#)
- [SKYNEWS INTL - 11300](#)
- [AL JAZEERA - 10929](#)
- [SKYNEWS INTL - 11594](#)
- [AL JAZEERA - 11579](#)
- [SKYNEWS INTL - 11827](#)

CNN INTERNATIONAL - 11553

<http://zattoo-production-zapl-sandbox.zattoo.com/watch/cnn-international/113875872/11553/1466568000000/1466571600000/01>

Language: en CID: cnn-international Retrieved Date: Wed Jun 22 06:00:00 CEST 2016



... Sarazen to . And then we subpoenaed The debate over whether this day only the EU's final campaigning ahead of Britain[United Kingdom]'s decision . And were preempted and then firing of more than about putting the region on hand . Big also Crockett Hillary now he's sitting back with their own nickname thinking of debt and Clinton is using from some business recall against . Hello everyone and what the viewers in the United States and around the world I never watch . And on John blows this is CNN news in July from attack This allows their campaigning before the UK[United Kingdom] is pivotal vote on whether to remain in the European Union ball . Suggested just too close to call just under 46 a half-a-million people have registered to vote a record . What where London[London] mayor Boris Johnson[Boris Johnson] leads the leave campaign and he story in a country wednesday trying to sway undecided voters

Figure 9.12.: Example of retrieved TV segments for query “英国 *boris johnson*”.

languages and modalities for a seamless semantic access to media streams. Firstly, it enables cross-lingual search in the sense that users can use keyword queries in any language (even in multiple languages) to retrieve multilingual media items. For this purpose, we use the multi-

lingual knowledge base as an interlingua to connect keyword queries and media items across languages. Through the semantic integration of different media streams, *XKnowSearch!* also supports cross-modal search by identifying a common model that suits the diversity of the data sources and combining the processed data with additional background knowledge. Some examples of the retrieved news articles, social media posts and TV segments for the keyword query “英国 *boris johson*” are shown in Figures 9.10, 9.11 and 9.12, respectively.

9.5. Related Work

We review the existing entity-based search systems and discuss their limitations, which serve as the motivation of our framework of cross-lingual semantic annotation and search.

EntEXPO (Liu et al., 2014) provides entity-based query expansion by finding a list of related entities of a single query entity and it allows users to manually adjust the weight of each related entity. However, there is no discussion about how to resolve the ambiguity of the query keywords and it does not concern with the queries containing concept names or multiple entities. EntEXPO seems to support search only in English.

Kuphi (Färber et al., 2014) employs semantic annotations of documents to enhance the performance of document retrieval. It allows interactive query reformulation by selecting the intended entity and adjusting the weights of related entities. However, the system assumes that a keyword query is a single entity name such that it cannot handle queries containing more than one entity name or concept names.

STICS (Hoffart et al., 2014b) has been proposed to support users in searching for terms, entities and categories. However, users have to specify the query entities and categories explicitly such that the ambiguity of queries can only be resolved by users. Moreover, it supports neither query expansion with related entities nor interactive query formulation / refinement. Finally, STICS also does not support cross-lingual search.

Recently, almost every major commercial Web search engine has announced their work on incorporating entity information from knowledge bases into its search process, including Google’s Knowledge Graph, Yahoo!’s Web of Objects and Microsoft’s Satori Graph / Bing Snapshots. However, there are still some limitations. Firstly, most search engines take into account only the most prominent entities matching the keyword query. Secondly, they can only understand individual entities, but cannot deal with a set of entities expressed by a concept name. Finally, they do not support cross-lingual search.

In summary, existing entity-based search systems cannot well address the challenges of *inflexibility*, *ambiguity* and *incompleteness*. More importantly, all of them do not support *cross-lingual search*. For example, EntEXPO seems to support only English and STICS supports both English and German, but neither of them can handle cross-lingual search. Although

Kuphi enables users to search documents in one language by using queries in another language, users have to specify the input language of the query, which can only be a single entity name. With the help of X-LiSA, XKnowSearch!, to the best of our knowledge, is the first entity-based system to multilingual and cross-lingual information retrieval with the goal of addressing these challenges, where users can issue keyword queries in any language, which can even contain keywords in multiple languages, for retrieving multilingual media items, especially in any other languages. In order to avoid the users' burden of specifying the query languages, we do not assume any input language given by users.

9.6. Conclusions

In this chapter, we demonstrated a framework that was built to break the barriers in between languages, channels and modalities for a seamless semantic access to media streams. Access is provided by multilingual keyword search, interactive entity search or SPARQL queries. News articles, social media posts and TV segments matching the query can be monitored live in a media stream.

Regarding future work, the described components are relying on explicit semantics only, restricting it to given entities in knowledge bases that can be annotated in text. Recent progress in cross-lingual and cross-modal representation learning enables a different retrieval approach that is not restricted to existing entities. Integrating those two approaches without losing the explainability of explicit semantics is a promising future research direction.

Part V.
Conclusions

10. Conclusions

We conclude this thesis by first summing up the research questions and our contributions. In addition, we provide an outlook on research directions for future work.

10.1. Summary

Given the multilingual Web as a repository of both documents and knowledge grounded in different languages, two major challenges for intelligent information access on the Web, namely semantic gap and language barrier, have been introduced in Chapter 1. In order to address these two challenges, we raised the following principal research question:

How to allow for semantic-aware and cross-lingual processing of Web documents and user queries by leveraging knowledge bases?

According to different tasks concerned in this thesis, namely semantic annotation and search as well as their cross-lingual extensions, the overall question is broken down into eight individual research questions for which our findings are summarized in the following:

Research Question 1. *How to enable context-aware and collective entity disambiguation for different types of input mentions in documents?*

In Chapter 3, we proposed a context-aware approach to collective entity disambiguation of the input mentions with different characteristics. The main contributions include the contextual entity detection based on a set of predefined part-of-speech (POS) tag patterns, which provides the context to help with entity disambiguation for the given input mentions, and the collective disambiguation using a class of algorithms for estimating the relative importance of candidate entities in the constructed disambiguation graph based on Markov chains.

Research Question 2. *How to enable salient entity discovery in documents?*

In Chapter 4, we aimed at Research Question 2 – targeting an approach to the new problem of salient entity linking. The main contribution lies in our proposed graph-based salient entity linking framework, which integrates several features including prior mention importance, mention-entity compatibility, entity-entity coherence and in particular a topic-sensitive model capturing entity-category association and document-specific category importance.

Research Question 3. *How to enable time-aware entity recommendation for temporal information needs?*

In Chapter 5, we presented a statistically sound probabilistic model that takes time-awareness into consideration for entity recommendation given temporal information needs. More specifically, we decomposed the task into several well defined probability distributions that reflect heterogeneous entity knowledge and show how all parameters of our probabilistic model can be effectively estimated solely based on data sources publicly available on the Web. Moreover, we have created new benchmark datasets to enable empirical evaluation of this new challenge.

Research Question 4. *How to enable effective and efficient keyword search on knowledge graphs?*

In Chapter 6, we aimed at a query rewriting solution to enable more effective and efficient keyword search on graph data. For this, we proposed a novel approach to probabilistic ranking and context-based computation of query rewrites. In addition, we investigated the impacts of our ranking mechanism and computation algorithm for query rewriting on both effectiveness and efficiency of keyword search, respectively. Moreover, we showed that the improvements on query rewriting achieved by our approach also carry over to the actual keyword search.

Research Question 5. *How to allow for an easy mapping of natural language expressions in different languages to entities in knowledge bases?*

In Chapter 7, we addressed Research Question 5 by constructing cross-lingual linked data lexica consisting of cross-lingual groundings of entities in knowledge bases. For this, we exploited various kinds of structures in Wikipedia, such as anchor text of hyperlinks and cross-language links, to derive different associations between natural language expressions extracted from Wikipedia editions in multiple languages and linked data resources.

Research Question 6. *How to enable cross-lingual keyword query interpretation?*

In Chapter 8, we proposed a knowledge base approach to cross-lingual query interpretation by transforming query keywords in different languages to their semantic representation, i.e., entities, to address the challenges that traditional keyword search systems mainly suffer from. The main contributions include a scoring mechanism for effective query interpretation ranking and a top-k graph exploration algorithm for efficient query interpretation generation.

Research Question 7. *How to enable cross-lingual entity linking in multilingual documents?*

With respect to Research Question 7, we presented a cross-lingual semantic annotation system in Chapter 9, which can link words or phrases in unstructured text in one language to entities in structured knowledge bases in any other language. We employed our cross-lingual linked data lexica for mention-entity matching and applied a concept-based approach for cross-lingual context similarity calculation. Moreover, our approach to collective entity disambiguation used by monolingual entity disambiguation has been adapted to this cross-lingual setting.

Research Question 8. *How to enable entity-based cross-lingual information retrieval (IR) by exploiting knowledge bases?*

Finally, we addressed Research Question 8 in Chapter 9. For this, we presented a novel system for entity-based cross-lingual IR, where users can issue keyword queries in any language, which can even contain keywords in multiple languages, for retrieving documents in any other languages. By leveraging entities in the multilingual knowledge base, keyword queries and Web documents in different languages are captured on their semantic level to address the ambiguity of terms and to bridge the language barrier between queries and documents.

10.2. Outlook

While we have addressed several key problems with regard to the overall research question in this work, there are still relevant directions for future work that need to be investigated. In the following, we briefly outline some further research questions on top of this thesis:

Future Work 1. *How to enable emerging entity discovery in documents?*

We presented our solutions to entity disambiguation and salient entity linking in Chapter 3 and Chapter 4, respectively. However, this work cannot well address the issue of *emerging entities*, namely entities referred to by mentions that are not contained in knowledge bases. Here, the key problem is to determine whether a mention refers to an existing entity in knowledge bases or represents a new entity. Traditional approaches addressed this problem by using a threshold of scores, i.e., when the score obtained by all candidate entities is too low, a mention is determined to represent a new entity. Recently, the work in (Hoffart et al., 2014a) proposed a new approach based on direct indication for a new entity by mining a corpus to extract possible representations of unknown entities. However, the problem of emerging entity discovery has not been well studied yet and there are still possibilities for improvement. The work in (Färber et al., 2016) thoroughly investigates all types of challenges that arise from out-of-knowledge-base entities for entity linking tasks.

Future Work 2. *How to enable joint entity linking and text categorization by exploiting knowledge bases?*

Wikipedia, as well as its derived knowledge bases like DBpedia and YAGO, has become a very important resource for performing a variety of text analysis tasks. Such knowledge bases contain a vast number of entities so that they can be used for the tasks of entity disambiguation and linking. On the other hand, page categorization in Wikipedia is a mixture of taxonomy and collaborative tagging such that the category hierarchy in the knowledge bases extracted from Wikipedia can be used for the identification of document topics (Schönhofen, 2009), also called text categorization. Currently, entity linking and text categorization based on knowledge bases are always performed separately. However, an entity in Wikipedia corresponding to one page is usually meaningfully categorized and the knowledge bases like DBpedia have

defined a proper ontology to reflect the relationships between entities and categories. By leveraging such relationships, a *joint optimization of entity linking and text categorization* could benefit the quality of both tasks.

Future Work 3. *How to enable semantic search given keyword queries with explicitly mentioned relations?*

In Chapter 9, we presented our system for entity-based cross-lingual semantic search, where both keyword queries and documents in different languages are represented as entities such that documents can be retrieved on the basis of relevance to entities instead of the term-based retrieval paradigm. However, our work does not support matching relations explicitly expressed in keyword queries against documents. In order to enable this type of semantic search, the following problems need to be investigated: (1) query interpretation needs to construct and rank the query graphs on the basis of both entities and relations mentioned in keyword queries; (2) documents have to be annotated with not only entities but also relations, which can be achieved by the recent advances in relation extraction as discussed in Chapter 2; and (3) documents should be retrieved on the basis of relevance to both entities and relations.

Future Work 4. *How to allow for the application of cross-lingual semantics without cross-language links in knowledge bases?*

In Chapter 7, we presented our approach that exploits cross-language links in Wikipedia for constructing the cross-lingual linked data lexica, which have been used by various tasks discussed in Chapter 8 and Chapter 9. Therefore, the performance of our work on cross-lingual processing depends on the cross-language link structure of Wikipedia, which should be consistent and needs to have enough coverage. However, the different Wikipedia language editions vary dramatically in how comprehensive they are such that only a small fraction of the sum of information that exists across all Wikipedias can be connected by cross-language links. Therefore, an open question remains about how to extract cross-lingual groundings of entities and relations in particular when cross-language links in knowledge bases are missing.

In summary, exploiting the semantics of information in multiple languages available on the Web is becoming an increasingly important issue for the new generation of intelligent applications. As we look at the future of semantic annotation and search, we believe that it is possible to not only benefit from the additional semantics of data but also contribute to building and using such semantics, especially in the multilingual and cross-lingual settings.

Bibliography

- Agirre, E. & Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 33–41.
- Alonso, O. & Zaragoza, H. (2008). Exploiting semantic annotations in information retrieval: ESAIR '08. *SIGIR Forum*, 42(1):55–58.
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. G. (2007). Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735.
- Baldwin, T. & Lui, M. (2010). Language identification: The long and the short of the matter. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*, pages 229–237.
- Balog, K., Dalton, J., Doucet, A., & Ibrahim, Y. (2016). Report on the eighth workshop on exploiting semantic annotations in information retrieval (ESAIR '15). *SIGIR Forum*, 50(1):49–57.
- Balog, K., Serdyukov, P., & de Vries, A. P. (2011). Overview of the TREC 2011 entity track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011.*
- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2670–2676.
- Bast, H., Buchhold, B., & Hausmann, E. (2016). Semantic search on text and knowledge bases. *Foundations and Trends in Information Retrieval*, 10(2-3):119–271.
- Bendersky, M. & Croft, W. B. (2008). Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 491–498.

BIBLIOGRAPHY

- Berberich, K., Bedathur, S. J., Neumann, T., & Weikum, G. (2007). A time machine for text search. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 519–526.
- Bergsma, S. & Wang, Q. I. (2007). Learning noun phrase query segmentation. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 819–826.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.
- Bi, B., Ma, H., Hsu, B. P., Chu, W., Wang, K., & Cho, J. (2015). Learning to recommend related entities to search users. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, China, February 2-6, 2015*, pages 139–148.
- Bizer, C., Heath, T., & Berners-Lee, T. (2009a). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009b). Dbpedia - a crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165.
- Blanco, R., Cambazoglu, B. B., Mika, P., & Torzec, N. (2013). Entity recommendations in web search. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, pages 33–48.
- Bollacker, K. D., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250.
- Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120.
- Bontcheva, K. & Cunningham, H. (2011). Semantic annotations and retrieval: Manual, semi-automatic, and automatic generation. In *Handbook of Semantic Web Technologies*, pages 77–116. Springer.
- Bontcheva, K., Tablan, V., & Cunningham, H. (2013). Semantic search over documents and ontologies. In *Bridging Between Information Retrieval and Databases - PROMISE Winter School 2013, Bressanone, Italy, February 4-8, 2013. Revised Tutorial Lectures*, volume 8173 of *Lecture Notes in Computer Science*, pages 31–53. Springer.

- Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117.
- Bron, M., Balog, K., & de Rijke, M. (2010). Ranking related entities: components and analyses. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1079–1088.
- Buitelaar, P. & Cimiano, P. (Eds.) (2014). *Towards the Multilingual Semantic Web, Principles, Methods and Applications*. Springer.
- Carlson, A., Betteridge, J., Wang, R. C., Jr., E. R. H., & Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 101–110.
- Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., & Trani, S. (2013). Dexter: an open source framework for entity linking. In *ESAIR'13, Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval, co-located with CIKM 2013, San Francisco, CA, USA, October 28, 2013*, pages 17–20.
- Cheng, G. & Qu, Y. (2009). Searching linked objects with falcons: Approach, implementation and evaluation. *Int. J. Semantic Web Inf. Syst.*, 5(3):49–70.
- Cheng, X. & Roth, D. (2013). Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1787–1796.
- Ciglan, M. & Nørvåg, K. (2010). Wikipop: personalized event detection system based on wikipedia page view statistics. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1931–1932.
- Cilibrasi, R. & Vitányi, P. M. B. (2007). The google similarity distance. *IEEE Trans. Knowl. Data Eng.*, 19(3):370–383.
- Coffman, J. & Weaver, A. C. (2010). A framework for evaluating database keyword search strategies. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 729–738.
- Coppersmith, D., Fleischer, L., & Rudra, A. (2010). Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Transactions on Algorithms*, 6(3):55:1–55:13.
- Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods*

BIBLIOGRAPHY

- in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716.
- Cucerzan, S. & Brill, E. (2004). Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 293–300.
- Dai, N. & Davison, B. D. (2010). Freshness matters: in flowers, food, and web authority. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 114–121.
- de Melo, G. & Weikum, G. (2009). Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 513–522.
- de Melo, G. & Weikum, G. (2010). Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1099–1108.
- Demidova, E., Zhou, X., & Nejdl, W. (2010). Iq^P: Incremental query construction, a probabilistic approach. In *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, pages 349–352.
- Demidova, E., Zhou, X., & Nejdl, W. (2012a). Freeq: an interactive query interface for freebase. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 325–328.
- Demidova, E., Zhou, X., & Nejdl, W. (2012b). A probabilistic scheme for keyword-based incremental query construction. *IEEE Trans. Knowl. Data Eng.*, 24(3):426–439.
- Demidova, E., Zhou, X., & Nejdl, W. (2013). Efficient query construction for large scale data. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 573–582.
- Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., & Zha, H. (2010). Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 331–340.
- Egozi, O., Markovitch, S., & Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. *ACM Trans. Inf. Syst.*, 29(2):8.
- Elsas, J. L. & Dumais, S. T. (2010). Leveraging temporal dynamics of document content in relevance ranking. In *Proceedings of the Third International Conference on Web Search*

- and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010, pages 1–10.
- Ermilov, I., Lehmann, J., Martin, M., & Auer, S. (2016). Lodstats: The data web census dataset. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, pages 38–46.
- Färber, M., Rettinger, A., & Asmar, B. E. (2016). On emerging entity detection. In *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, pages 223–238.
- Färber, M., Zhang, L., & Rettinger, A. (2014). Kuphi - an investigation tool for searching for and via semantic relations. In *The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 349–354.
- Fellbaum, C. (Ed.) (1998). *WordNet: an electronic lexical database*. MIT Press.
- Ferragina, P. & Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75.
- Finkel, J. R., Grenager, T., & Manning, C. D. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*.
- Fischer, L., Blanco, R., Mika, P., & Bernstein, A. (2015). Timely semantics: A study of a stream-based ranking system for entity relationships. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, pages 429–445.
- Gabrilovich, E. & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1606–1611.
- Gamon, M., Yano, T., Song, X., Apacible, J., & Pantel, P. (2013). Identifying salient entities in web pages. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2375–2380.
- Gao, L., Yu, X., & Liu, Y. (2011). Keyword query cleaning with query logs. In *Web-Age Information Management - 12th International Conference, WAIM 2011, Wuhan, China, September 14-16, 2011. Proceedings*, pages 31–42.
- Giger, H. (1988). Concept based retrieval in classical IR systems. In *SIGIR'88, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, June 13-15, 1988*, pages 275–289.

BIBLIOGRAPHY

- Han, X., Sun, L., & Zhao, J. (2011). Collective entity linking in web text: a graph-based method. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 765–774.
- Hassan, S. & Mihalcea, R. (2011). Semantic relatedness using salient semantic analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*.
- Hastie, T. & Tibshirani, R. (1997). Classification by pairwise coupling. In *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*, pages 507–513.
- Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii*, pages 517–526.
- Haveliwala, T. H. (2003). Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.*, 15(4):784–796.
- He, H., Wang, H., Yang, J., & Yu, P. S. (2007). Blinks: ranked keyword searches on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*, pages 305–316.
- Heath, T. & Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers.
- Hoffart, J., Altun, Y., & Weikum, G. (2014a). Discovering emerging entities with ambiguous names. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 385–396.
- Hoffart, J., Milchevski, D., & Weikum, G. (2014b). STICS: searching with strings, things, and cats. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast, QLD, Australia - July 06 - 11, 2014*, pages 1247–1248.
- Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., & Weikum, G. (2012). KORE: keyphrase overlap relatedness for entity disambiguation. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 545–554.
- Hoffart, J., Yosef, M. A., Bordino, I., Fürstenaу, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., & Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792.

- Hristidis, V., Gravano, L., & Papakonstantinou, Y. (2003). Efficient ir-style keyword search over relational databases. In *VLDB*, pages 850–861.
- Irmak, U., von Brzeski, V., & Kraft, R. (2009). Contextual ranking of keywords using click data. In *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, pages 457–468.
- Järvelin, K. & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *SIGIR*, pages 41–48.
- Jeh, G. & Widom, J. (2003). Scaling personalized web search. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 271–279.
- Jones, R., Rey, B., Madani, O., & Greiner, W. (2006). Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 387–396.
- Jurafsky, D. & Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., & Karambelkar, H. (2005). Bidirectional expansion for keyword search on graph databases. In *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, pages 505–516.
- Kang, C., Vadrevu, S., Zhang, R., van Zwol, R., Pueyo, L. G., Torzec, N., He, J., & Chang, Y. (2011). Ranking related entities for web search queries. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 67–68.
- Kanhabua, N., Blanco, R., & Nørsvåg, K. (2015). Temporal information retrieval. *Foundations and Trends in Information Retrieval*, 9(2):91–208.
- Kanhabua, N. & Nørsvåg, K. (2010). Determining time of queries for re-ranking search results. In *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, pages 261–272.
- Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's smo algorithm for svm classifier design. *Neural Comput.*, 13(3):637–649.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.
- Kulkarni, A., Teevan, J., Svore, K. M., & Dumais, S. T. (2011). Understanding temporal query dynamics. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 167–176.
- Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International*

BIBLIOGRAPHY

- Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 457–466.
- Ladwig, G. & Tran, T. (2011). Index structures and top-k join algorithms for native keyword search databases. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1505–1514.
- Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):32–38.
- Li, G., Ooi, B. C., Feng, J., Wang, J., & Zhou, L. (2008). Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 903–914.
- Li, M., Zhu, M., Zhang, Y., & Zhou, M. (2006). Exploring distributional similarity based models for query spelling correction. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*.
- Li, X. & Croft, W. B. (2003). Time-based language models. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*, pages 469–475.
- Liu, F., Yu, C. T., Meng, W., & Chowdhury, A. (2006). Effective keyword search in relational databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29, 2006*, pages 563–574.
- Liu, X., Yang, P., & Fang, H. (2014). Entexpo: An interactive search system for entity-bearing queries. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 784–788.
- Lu, Y., Wang, W., Li, J., & Liu, C. (2011). Xclean: Providing valid spelling suggestions for xml keyword queries. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, pages 661–672.
- Luo, Y., Lin, X., Wang, W., & Zhou, X. (2007). Spark: top-k keyword query in relational databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, June 12-14, 2007*, pages 115–126.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- Mays, E., Damerau, F. J., & Mercer, R. L. (1991). Context based spelling correction. *Inf. Process. Manage.*, 27:517–522.

- Mendes, P. N., Jakob, M., & Bizer, C. (2012). Dbpedia: A multilingual cross-domain knowledge base. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 1813–1817.
- Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, pages 1–8.
- Mihalcea, R. & Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 233–242.
- Milne, D. & Witten, I. H. (2008a). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30. AAAI Press.
- Milne, D. N. & Witten, I. H. (2008b). Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 509–518.
- Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *TACL*, 2:231–244.
- Nastase, V., Strube, M., Boerschinger, B., Zirn, C., & Elghafari, A. (2010). Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*.
- Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).
- Navigli, R. (2012). A quick tour of word sense disambiguation, induction and related approaches. In *SOFSEM 2012: Theory and Practice of Computer Science - 38th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 21-27, 2012. Proceedings*, pages 115–129.
- Navigli, R. & Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.
- Nørvåg, K. (2004). Supporting temporal text-containment queries in temporal document databases. *Data Knowl. Eng.*, 49(1):105–125.
- Osborne, M., Petrovic, S., McCreddie, R., Macdonald, C., & Ounis, I. Bieber no more: First Story Detection using Twitter and Wikipedia. *SIGIR 2012 Workshop on Time-aware Information Access (TAIA2012)*.

BIBLIOGRAPHY

- Paranjpe, D. (2009). Learning document aboutness from implicit user feedback and document structure. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 365–374.
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508.
- Piccinno, F. & Ferragina, P. (2014). From tagme to WAT: a new entity annotator. In *ERD'14, Proceedings of the First ACM International Workshop on Entity Recognition & Disambiguation, July 11, 2014, Gold Coast, Queensland, Australia*, pages 55–62.
- Platt, J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA.
- Potthast, M., Stein, B., & Anderka, M. (2008). A wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval, 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*, pages 522–530.
- Pound, J., Mika, P., & Zaragoza, H. (2010). Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 771–780.
- Pu, K. Q. & Yu, X. (2008). Keyword query cleaning. *PVLDB*, 1(1):909–920.
- Qiu, Y. & Frei, H. (1993). Concept based query expansion. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 160–169.
- Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1375–1384.
- Rizzo, G., van Erp, M., & Troncy, R. (2014). Benchmarking the extraction and disambiguation of named entities on the semantic web. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 4593–4600.
- Röder, M., Usbeck, R., Hellmann, S., Gerber, D., & Both, A. (2014). N³ - A collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 3529–3533.
- Sarawagi, S. (2008). Information extraction. *Foundations and Trends in Databases*, 1(3):261–377.
- Schönhofen, P. (2009). Identifying document topics using the wikipedia category network. *Web Intelligence and Agent Systems*, 7(2):195–207.

- Seneta, E. (2006). *Non-negative matrices and Markov chains; rev. version*. Springer series in statistics. Springer, New York, NY.
- Shen, W., Wang, J., & Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460.
- Shen, W., Wang, J., Luo, P., & Wang, M. (2012). LINDEN: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 449–458.
- Shokouhi, M. & Radinsky, K. (2012). Time-sensitive query auto-completion. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 601–610.
- Singh, S., Subramanya, A., Pereira, F., & McCallum, A. (2012). Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015.
- Sorg, P. & Cimiano, P. (2008). Cross-language information retrieval with explicit semantic analysis. In *Working Notes for CLEF 2008 Workshop co-located with the 12th European Conference on Digital Libraries (ECDL 2008), Aarhus, Denmark, September 17-19, 2008*.
- Spitkovsky, V. I. & Chang, A. X. (2012). A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 3168–3175.
- Steinmetz, N., Knuth, M., & Sack, H. (2013). Statistical analyses of named entity disambiguation benchmarks. In *Proceedings of the NLP & DBpedia workshop co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 22, 2013*.
- Steinmetz, N. & Sack, H. (2013). Semantic multimedia information retrieval based on contextual descriptions. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, pages 382–396.
- Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706.
- Tan, B. & Peng, F. (2008). Unsupervised query segmentation using generative language models and wikipedia. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 347–356, New York, NY, USA. ACM.
- Trampuš, M. & Novak, B. (2012). Internals of an aggregated web news feed. In *Proceedings*

BIBLIOGRAPHY

- of the Conference on Data Mining and Data Warehouses (SiKDD 2012) co-located with the 15th International Multiconference on Information Society*, pages 431–434.
- Tran, T., Cimiano, P., Rudolph, S., & Studer, R. (2007). Ontology-based interpretation of keywords for semantic search. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 523–536.
- Tran, T., Herzig, D. M., & Ladwig, G. (2011). Semsearchpro - using semantics throughout the search process. *J. Web Sem.*, 9(4):349–364.
- Tran, T., Wang, H., Rudolph, S., & Cimiano, P. (2009). Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, pages 405–416.
- Tran, T. & Zhang, L. (2014). Keyword query routing. *IEEE Trans. Knowl. Data Eng.*, 26(2):363–375.
- Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., & Decker, S. (2010). Sig.ma: Live views on the web of data. *J. Web Sem.*, 8(4):355–364.
- Usbeck, R., Ngomo, A. N., Röder, M., Gerber, D., Coelho, S. A., Auer, S., & Both, A. (2014). AGDISTIS - graph-based disambiguation of named entities using linked data. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 457–471.
- Usbeck, R., Röder, M., Ngomo, A. N., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Waitelonis, J., & Wesemann, L. (2015). GERBIL: general entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1133–1143.
- van Erp, M., Rizzo, G., & Troncy, R. (2013). Learning with the web: Spotting named entities on the intersection of NERD and machine learning. In *Proceedings of the Concept Extraction Challenge at the Workshop on 'Making Sense of Microposts', Rio de Janeiro, Brazil, May 13, 2013*, pages 27–30.
- van Zwol, R., Pueyo, L. G., Muralidharan, M., & Sigurbjörnsson, B. (2010). Machine learned ranking of entity facets. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 879–880.
- Voorhees, E. M. (1993). Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 171–180.

- Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 61–69.
- Vrandečić, D. & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.
- Wei, X. & Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 178–185.
- White, S. & Smyth, P. (2003). Algorithms for estimating relative importance in networks. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 266–275.
- Yao, J., Cui, B., Hua, L., & Huang, Y. (2012). Keyword query reformulation on structured data. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 953–964.
- Yih, W., Goodman, J., & Carvalho, V. R. (2006). Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 213–222.
- Yu, X., Ma, H., Hsu, B. P., & Han, J. (2014). On building entity recommender systems using user click log and freebase knowledge. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 263–272.
- Zenz, G., Zhou, X., Minack, E., Siberski, W., & Nejdil, W. (2009). From keywords to semantic queries - incremental query construction on the semantic web. *J. Web Sem.*, 7(3):166–176.
- Zhang, L., Dong, Y., & Rettinger, A. (2015a). Towards entity correctness, completeness and emergence for entity recognition. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 143–144.
- Zhang, L., Färber, M., & Rettinger, A. (2014a). xlid-lexica: Cross-lingual linked data lexica. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 2101–2105.
- Zhang, L., Färber, M., & Rettinger, A. (2016a). Xknowsearch!: Exploiting knowledge bases for entity-based cross-lingual information retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 2425–2428.

BIBLIOGRAPHY

- Zhang, L., Liu, C., & Rettinger, A. (2015b). A topic-sensitive model for salient entity linking. In *Proceedings of the Third International Workshop on Linked Data for Information Extraction (LD4IE2015) co-located with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pennsylvania, USA, October 12, 2015.*, pages 63–70.
- Zhang, L. & Rettinger, A. (2014). X-lisa: Cross-lingual semantic annotation. *PVLDB*, 7(13):1693–1696.
- Zhang, L., Rettinger, A., & Philipp, P. (2016b). Context-aware entity disambiguation in text using markov chains. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016*, pages 49–56.
- Zhang, L., Rettinger, A., & Thoma, S. (2014b). Bridging the gap between cross-lingual nlp and dbpedia by exploiting wikipedia. In *Proceedings of the Second NLP&DBpedia Workshop (NLP & DBpedia 2014) co-located the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*
- Zhang, L., Rettinger, A., & Zhang, J. (2016c). A knowledge base approach to cross-lingual keyword query interpretation. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 615–631.
- Zhang, L., Rettinger, A., & Zhang, J. (2016d). A probabilistic model for time-aware entity recommendation. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I*, pages 598–614.
- Zhang, L., Thalhammer, A., Rettinger, A., Färber, M., Mogadala, A., & Denaux, R. (2017). The xlime system: Cross-lingual and cross-modal semantic annotation, search and recommendation over live-tv, news and social media streams. *J. Web Sem.*, 46:20–30.
- Zhang, L., Tran, T., & Rettinger, A. (2013). Probabilistic query rewriting for efficient and effective keyword search on graph data. *PVLDB*, 6(14):1642–1653.
- Zhang, L., Tran, T., & Rettinger, A. (2015c). A theoretical analysis of cross-lingual semantic relatedness in vector space models. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR 2015, Northampton, Massachusetts, USA, September 27-30, 2015*, pages 241–250. ACM.