EXPLOITING ENTITIES FOR QUERY EXPANSION

WLADMIR CARDOSO BRANDÃO

EXPLOITING ENTITIES FOR QUERY EXPANSION

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

ADVISOR: NIVIO ZIVIANI

Belo Horizonte

October 2013

Brandão, Wladmir Cardoso

B817v Exploiting Entities for Query Expansion / Wladmir Cardoso Brandão. — Belo Horizonte, 2013
xvi, 93 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação
Orientador: Nivio Ziviani

1. Computação - Teses. 2. Sistemas de recuperação da informaçcão - Teses. 3. Aprendizado do computador - Teses. 4. Wikipedia - Teses. I. Orientador. II. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Exploiting entities for query expansion

WLADMIR CARDOSO BRANDÃO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. NIVIO ZIVIANI - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ADRIANO ALONSO VELOSO
Departamento de Ciência da Computação - UFMG

PROF. EDLENO SILVA DE MOURA
Departamento de Ciência da Computação - UFAM

PROF. MARIANO P. CONSENS
University of Toronto

PROF. RODRYGO LUIS TEODORO SANTOS
Departamento de Ciência da Computação - UFMG

PROFA. VIVIANE PEREIRA MOREIRA
Departamento de Informática Aplicada - UFRGS

Belo Horizonte, 18 de novembro de 2013.

To my son João Pedro, the reason for all.
From the beginning, along the path, until the end.

Acknowledgments

I would like to express my deepest gratitude to several people for their immense support during the course of my PhD. First and foremost, I am extremely grateful to my family, specially my parents Waldemar de Oliva Brandão and Telma Cardozo Brandão for their continuous and unconditional love, for always believing in me, and for their support in my decisions. Without them I could not have made it here.

I also would like to express my sincere gratitude to my advisor Prof. Nivio Ziviani. His entrepreneurship, experience, sincerity, critical view, and focus on results helped shape this thesis, and contributed considerably to my path towards an academic and research career. I am also grateful to Altigran S. da Silva, Edleno S. de Moura, and Rodrygo L. T. Santos for the significant contributions to the development of this thesis, and for collaborating in various other research endeavours.

I thank my labmates at the LATIN and LBD laboratories: Humberto Mossri de Almeida, Evandrino Gomes Barros, Denilson Pereira, Fabiano Botelho, Anisio Lacerda, Thierson Rosa, Guilherme Menezes, Alan Castro, Wallace Favoreto, Rickson Guidolini, Osvaldo Matos, Sabir Ribas, Thales Costa, Adolfo Guimarães, Aécio Santos, Itamar Hata, Aline Bessa, Cristiano Carvalho, Vitor Campos de Oliveira, and Vinicius Tinti, for the stimulating discussions, for the sleepless nights we were working before deadlines, and for all the fun we have had in the last five years. Also, I thank my colleagues in Federal University of Amazonas: Klessius Berlt, André Carvalho, and Karane Vieira.

I must also thank the administrative staff of the Computer Science Department of the Federal University of Minas Gerais, by attentive and caring support and for efficiently meeting my demands, specially Lizete Paula, Murilo Monteiro, Sônia Vaz de Melo, Renata Rocha, Laszlo Pinto, Heitor Motta, and Rosencler de Oliveira.

Lastly, I thank the partial support given by the Brazilian government, particularly the Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6), Project InfoWeb (grant MCT/CNPq/CT-INFO 550874/2007-0), and Project MinGroup (grant CNPq/CT-Amazônia 575553/2008-1).

"Science is a way of thinking much more than it is a body of knowledge."
(Carl Sagan)

Abstract

A substantial fraction of web search queries contain references to entities, such as persons, organizations, and locations. This significant presence of named entities in queries provides an opportunity for web search engines to improve their understanding of the user's information need. In this work, we investigate the entity-oriented query expansion process. Particularly, we propose two novel and effective query expansion approaches that exploit semantic sources of evidence to devise discriminative term features, and machine learning techniques to effectively combine these features in order to rank candidate expansion terms. As a result, not only do we select effective expansion terms, but we also weigh these terms according to their predicted effectiveness. In addition, since our query expansion approaches consider Wikipedia infoboxes as a source of candidate expansion terms, a frequent obstacle is that only about 20% of Wikipedia articles have an infobox. To overcome this problem we propose WAVE, a self-supervised approach to autonomously generate infoboxes for Wikipedia articles.

First, we propose UQEE, an unsupervised entity-oriented query expansion approach, which effectively selects expansion terms using taxonomic features devised by the semantic structure implicitly provided by infobox templates. We show that query expansion using infoboxes presents a better trade-off between retrieval performance and query latency. Moreover, we demonstrate that the automatically generated infoboxes provided by WAVE are as effective as manually generated infoboxes for query expansion. Lastly, we propose L2EE, a learning to rank approach for entity-oriented query expansion, which considers semantic evidence encoded in the content of Wikipedia article fields, and automatically labels training examples proportionally to their observed retrieval effectiveness. Experiments on three TREC web test collections attest the effectiveness of L2EE, with significant gains compared to UQEE and state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches.

Keywords: Query expansion, relevance feedback, machine learning, learning to rank, named entity recognition, Wikipedia, infobox.

Contents

Acknowledgments **ix**

Abstract **xiii**

1 Introduction **1**
1.1 Thesis Statement . . . 2
1.2 Thesis Contributions . . . 2
1.3 Origins of the Material . . . 3
1.4 Thesis Outline . . . 4

2 Web Search **7**
2.1 Search Engines . . . 9
2.1.1 Crawling . . . 9
2.1.2 Indexing . . . 11
2.1.3 Querying . . . 13
2.2 Query Understanding . . . 14
2.2.1 Feedback Methods . . . 17
2.2.2 Query Expansion . . . 18
2.2.3 Related Approaches . . . 23
2.3 Summary . . . 25

3 Unsupervised Entity-Oriented Query Expansion **27**
3.1 Overview . . . 28
3.2 Entity Representation . . . 29
3.3 Entity Resolution . . . 31
3.4 Ranking Entity Terms . . . 33
3.4.1 Ranking Features . . . 33
3.4.2 Combining Rankings . . . 34
3.5 Experiments . . . 35

3.5.1 Setup . . . 35
3.5.2 Results . . . 37
3.6 Summary . . . 40

4 Automatic Infobox Generation **43**
4.1 Overview . . . 44
4.2 Processing Wikipedia Corpus . . . 46
4.3 Classifying Articles and Sentences . . . 47
4.3.1 Article Classifier . . . 47
4.3.2 Sentence Classifier . . . 47
4.4 Filtering Sentences . . . 48
4.5 Extracting Values for Attributes . . . 48
4.5.1 Window-Based Segmentation . . . 48
4.5.2 CRF Extractor . . . 49
4.6 Experiments . . . 50
4.6.1 Setup . . . 50
4.6.2 Results . . . 52
4.7 Summary . . . 53

5 Supervised Entity-Oriented Query Expansion **55**
5.1 Overview . . . 56
5.2 Entity Representation and Resolution . . . 57
5.3 Ranking Entity Terms . . . 57
5.3.1 Learning a Ranking Model . . . 57
5.3.2 Ranking Features . . . 59
5.4 Experiments . . . 60
5.4.1 Setup . . . 60
5.4.2 Results . . . 63
5.5 Summary . . . 71

6 Conclusions and Future Work **73**
6.1 Summary of Contributions . . . 74
6.2 Summary of Conclusions . . . 75
6.3 Directions for Future Research . . . 77
6.4 Final Remarks . . . 80

Bibliography **81**

Chapter 1

Introduction

Search engines have become the primary gateway for finding information on the Web. The leading web search engine has recently reported to be answering a total of 100 billion queries each month, and to be tracking over 30 trillion unique URLs [Cutts, 2012]. Given the size of the Web and the short length of typical web search queries [Jansen et al., 2000; Gabrilovich et al., 2009], there may be billions of pages matching a single query. In this context, an improved understanding of the information need underlying the user's query becomes paramount for an improved search experience. Indeed, misinterpreting this query may result in relevant documents never being retrieved, regardless of how sophisticated the subsequent ranking process is [Li, 2010].

A particularly effective query understanding operation is query expansion [Rocchio, 1971; Lavrenko and Croft, 2001; Zhai and Lafferty, 2001b]. Given a set of feedback documents, such an approach appends additional terms to the query in order to close the gap between the vocabulary of the query and that of potentially relevant documents. Automatic query expansion is typically performed in the context of pseudo-relevance feedback, in which case the feedback set comprises the top retrieved documents for the query, which are assumed to be relevant [Rocchio, 1971]. However, this assumption is often invalid, which has prompted the development of improved mechanisms for selecting effective feedback documents [He and Ounis, 2009], and effective expansion terms [Cao et al., 2008; Lee et al., 2009; Udupa et al., 2009].

Recently, entity-oriented pseudo-relevance feedback approaches that exploit named entities have also been shown to be effective for query expansion [Xu et al., 2009]. In particular, queries with named entities, such as persons, organizations, and locations, account for over 70% of the web search traffic [Guo et al., 2009]. Such queries offer a unique opportunity to use knowledge bases as repositories of high-quality feed-

back documents in order to improve the understanding of the user's information need.

The main objective of this work is to investigate the entity-oriented query expansion problem. For this purpose, novel entity-oriented query expansion approaches are proposed and compared with state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches reported in the literature.

1.1 Thesis Statement

The statement of this thesis is that the use of multiple sources of semantic evidence on entities to devise discriminative term features, as well as the use of machine learning techniques to combine these features to rank candidate expansion terms are effective for query expansion.

1.2 Thesis Contributions

The key contributions of this thesis can be summarised as follows:

1. We exploit previously underexploited sources of semantic evidence on entities from a knowledge base as features to rank candidate expansion terms.

   Typically, query expansion approaches select candidate expansion terms from a set of feedback documents assumed to be relevant and, more recently, from a set of feedback *entities* from external knowledge bases, using the term frequency as a feature to rank these candidates. In this thesis, we show that the proximity of the candidate term to the original query terms, its frequency across multiple entity fields (e.g., title, summary, body from related articles), as well as in category descriptors are effective features for query expansion. Moreover, we show that Wikipedia infoboxes[1] are important sources of effective expansion terms. To this end, we derive an entity repository from Wikipedia, using the aforementioned features to select, from the infobox field related to an entity recognized in a query, the candidate expansion terms for the query.

2. We introduce and evaluate an approach to automatically generate infoboxes for Wikipedia articles.

   Only about 20% of Wikipedia articles have an infobox, which is an obstacle to use infobox descriptors for query expansion. In this thesis, we introduce

[1] Infoboxes are special tabular structures that present a set of attribute-value pairs describing different aspects of Wikipedia articles.

WAVE (Wikipedia Attribute-Value Extractor), a self-supervised approach to autonomously extract attribute-value pairs from the content of Wikipedia articles. We also evaluate its effectiveness to automatically generate infoboxes, by showing that it significantly outperforms the state-of-art approach in the literature. Moreover, the unsupervised entity-oriented query expansion approach presented in Chapter 3 uses WAVE to automatically generate infoboxes, which are shown to be as effective as manually generated infoboxes.

3. We introduce two novel and effective entity-oriented query expansion approaches.

   The entity-oriented pseudo-relevance feedback approaches reported in the literature typically leverage one or more external knowledge bases, such as Wikipedia and ConceptNet, as a repository of feedback documents. In this thesis, we introduce two novel approaches to address the entity-oriented query expansion problem. UQEE (Unsupervised Query Expansion using Entities) is an unsupervised approach that leverages well-known features, adapting them to deal properly with entities, ultimately improving the accuracy in selecting effective expansion terms. L2EE (Learning to Expand using Entities) is a supervised approach that tackles query expansion as a learning to rank problem. As a result, not only does it select effective expansion terms, but it also weighs these terms according to their predicted effectiveness when added to the query. In addition, these approaches are used to assess the effectiveness of sources of semantic evidence from Wikipedia, including infoboxes, for query expansion.

4. We thoroughly evaluate the proposed approaches and their impact on web search effectiveness using standard TREC web test collections.

   Our thorough experiments validate the aforementioned contributions in comparison to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches from the literature. Additionally, we meticulously investigate the robustness of our supervised approach when applied for queries with little room for improvement, or when no Wikipedia pages are considered in the search results. Moreover, we show that statistical and proximity features are particularly suitable for selecting effective expansion terms.

1.3 Origins of the Material

Most of the material presented in this thesis has previously been published in journal and conference papers. We now give an overview of published research:

- Chapter 3 describes UQEE, an unsupervised entity-oriented query expansion approach proposed by Brandão et al. [2011], presenting the category descriptors devised from Wikipedia infobox templates, as well as effective entity features. In addition, it empirically validates the proposed approach in contrast to a standard retrieval baseline.

- Chapter 4 provides motivations for automatic generation of Wikipedia infoboxes, and describes WAVE, a self-supervised approach to autonomously extract attribute-value pairs from Wikipedia articles proposed by Brandão et al. [2010]. Moreover, it presents experimental results to validate the proposed approach in contrast to a state-of-the-art approach from the literature.

- Chapter 5 extends the investigations by Brandão et al. [2011] on sources of semantic evidence for query expansion, and describes L2EE, a supervised learning entity-oriented query expansion approach proposed by Brandão et al. [2013]. Additionally, it validates the proposed approach in contrast to the current state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches, as well as to the unsupervised approach proposed by Brandão et al. [2011].

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 reviews the related literature on web search. Particularly, it begins by describing web information retrieval and search engine architecture and components. The role of query representation and understanding on web search is then discussed. Finally, the chapter ends with a discussion about query expansion, presenting traditional approaches and state-of-the-art baselines in the literature, as well as the typical entity-oriented query expansion process, ultimately laying the foundations for the several experiments conducted in this thesis.

- Chapter 3 introduces UQEE, an unsupervised approach for entity-oriented query expansion, and reports on its experimental evaluation. First, the off-line and on-line query processing stages are described, the entity representation and entity resolution steps required to recognize entities in queries are presented, and the procedure used to rank candidate expansion terms is described. Next, the experimental methodology that serves as the basis for the experiments is described.

Lastly, UQEE is thoroughly validated in comparison to a standard retrieval baseline.

- Chapter 4 describes WAVE, a self-supervised approach to autonomously extract attribute-value pairs from Wikipedia articles, as well as the setup and results of the experimental evaluation of the approach. In particular, the chapter begins by discussing the need of an approach to automatically generate Wikipedia infoboxes in order to support our entity-oriented query expansion approaches. Next, the components of WAVE are described. Lastly, WAVE is validated in comparison to the state-of-the-art approach reported in the literature.

- Chapter 5 introduces L2EE, a supervised approach for entity-oriented query expansion, and reports on its experimental evaluation. The chapter begins by describing the off-line and on-line query processing stages, focusing on the differences from UQEE. Their commonalities in entity representation are then discussed, and the procedure to learn a ranking model, as well as the features used by the model are described. Next, the experimental methodology that serves as the basis for the experiments is described. Finally, L2EE is thoroughly validated in comparison to UQEE, and effective state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches from the literature.

- Chapter 6 provides a summary of the contributions and the conclusions made throughout the chapters, and presents directions for future research.

Chapter 2

Web Search

People typically search information sources looking for answers to their practical questions, such as *"Who is this person?"*, *"Where is this place?"*, *"What does this company do?"*, and *"How can I accomplish this task?"*. They consume information to fill up knowledge gaps in order to solve practical problems [Meadow and Yuan, 1997]. In short, questions arising from practical problems create information needs. Due to the diversity and size of today's information sources, retrieving information from such sources can be an exhausting experience. Thus, the use of systems that support an effective information retrieval becomes paramount to fulfil people's information needs.

According to Baeza-Yates and Ribeiro-Neto [2011], information retrieval deals with the representation, storage, organization of, and access to information items in order to provide the users with easy access to information of their interest. Considering textual information retrieval, information items typically correspond to documents, while information needs are translated into natural language queries. Indeed, an information retrieval system must retrieve all relevant documents related to a user query while retrieving as few non-relevant documents as possible in order to accomplish the user's information needs. The challenge is not only to decide which documents to consider, or how to extract information from such considered documents, or even how to express information needs as queries, but mostly to decide what is relevant for users.

In the end of the 1980's, Tim Berners-Lee conceived the World Wide Web, or simply the Web [Berners-Lee, 1989]. From there, the Web has increasingly become a very large, public, and unstructured repository of multimedia data, ranging from text to images, audio and video, encoded in different formats and multiple languages. A recent report revealed that the Web comprises over 30 trillion uniquely addressable documents [Cutts, 2012]. Furthermore, documents in the Web are interconnected by hyperlinks, making the Web a huge information network where documents are

represented as nodes, and hyperlinks are represented as directed edges between documents [Kleinberg et al., 1999; Broder et al., 2000]. The massive-scale, heterogeneous, and interconnected nature of the Web trigger the need for efficient tools to manage, retrieve and filter its informational content [Baeza-Yates and Ribeiro-Neto, 2011]. In this environment, deciding which documents to consider, how to extract information from such considered documents, how to express information needs as queries, and what is relevant for users becomes an even greater challenge.

Search engines are information retrieval systems that model the Web as a full-text data repository, where all query processing must be done without accessing the source of the documents [Baeza-Yates and Ribeiro-Neto, 2011]. They have become the primary gateway for finding information on the Web. The leading commercial search engine reported the conduction of more than 100 billion searches each month, and over 3.3 billion searches each day [Cutts, 2012]. To effectivelly deal with this massive volume of search requests, a competitive search engine has to collect a significant set of documents from the Web, extract from them information that potentially interest the users, provide an interface to receive user queries, interpret such queries assuming the user intention, and finally match queries and documents producing a ranking of relevant documents that meet the user information needs.

A particularly challenging information retrieval problem is to identify the underlying search intent[1] on a particular query representation, a problem known as query understanding [Croft et al., 2011]. Typical query understanding operations include refinements of the original query, such as spelling correction [Li et al., 2006], acronym expansion [Jain et al., 2007], stemming [Porter, 1980; Peng et al., 2007], query reduction [Kumaran and Allan, 2008], query segmentation [Risvik et al., 2003], and query classification [Beitzel et al., 2005].

The primary application of interest for this thesis is query expansion, a more sophisticated query understanding operation [Lavrenko and Croft, 2001; Rocchio, 1971; Zhai and Lafferty, 2001b]. Therefore, our discussion is not aimed at providing an exhaustive survey on web search. Instead, it covers selected bibliography which is broad enough to contextualize the query expansion environment. With this in mind, Section 2.1 describes the basic retrieval process of a search engine, introducing the main components and activities in this process. Additionally, Section 2.2 describes the query representation and understanding process in a web search setting, discussing feedback methods, and presenting current approaches in the literature for query expansion.

[1]Search intent, query intent and information need are synonyms. The notion of an information need or problem underlying a query has been discussed in the IR literature for many years, and it was generally agreed that search intent and query intent is another way of referring to the same idea.

2.1 Search Engines

Nowadays, search engines are the primary gateway for finding information on the Web. Briefly, they receive user's queries using a simple web interface and process them, ultimately retrieving an ordered list of documents which potentially interest the users. For this, they gather information from the Web, and store it into repositories, creating from this point an index structure for fast retrieving and ranking of documents. Figure 2.1 shows the main components and tasks behind the retrieval flow of current search engines.

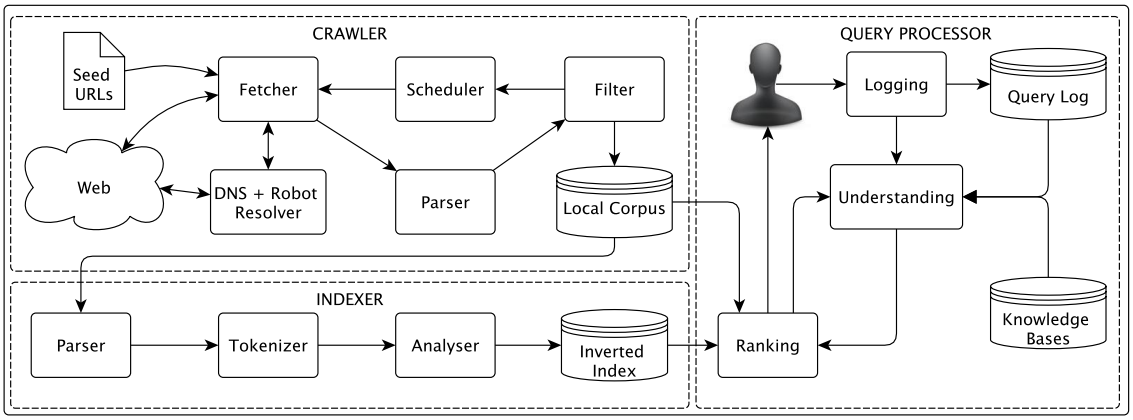

Figure 2.1. The components and tasks of current search engines.

As illustrated in Figure 2.1, search engines are composed by a crawler component, responsible for gathering information from the Web to build the local corpus of documents, an indexer component, responsible for creating the inverted index from the local corpus, and a query processor component, which handles the user's requests in order to produce and deliver a ranking of relevant documents. In the following, we describe in more details each one of these components. Particularly, Section 2.1.1 describes the crawling process triggered by the crawler, Section 2.1.2 describes the indexing process triggered by the indexer, and Section 2.1.3 describes the querying process triggered by the query processor. In each one of these sections, some challenges, trends and research issues on the respective process are also presented.

2.1.1 Crawling

As illustrated in Figure 2.1, the crawling process is performed by the crawler component. This process consists of collecting documents from the Web as fast as possible to build a comprehensive local corpus of documents, later used for indexing and searching [Pant et al., 2004]. For this, the crawler sends requests for documents to web

servers, processing the responses in order to download and store the collected documents into the corpus.

Particularly, in the beginning the *fetcher* downloads documents from web servers by processing a seed of URLs.[2] However, before the *fetcher* sends the request, the *DNS + Robots resolver* must translate the URL domain into an IP address, further checking if the politeness policy of the web server allows the document download. Next, the downloaded documents are parsed to extract text and internal URLs to continuously feedback the crawler. The parsed content is used by the *filter* to decide if an extracted URL must be considered, and if the content of the downloaded document must be actually stored, since there are documents without interesting information for the users, such as web spam. Further in this section we present the challenging problem of avoiding web spam in search. Finally, the *scheduler* merges visited and not visited URLs in a unique ordered queue to be consumed by the *fetcher*, and a new crawling cycle begins. The scheduling task is crucial in crawlers, since quality and freshness estimations, as well as URL rearrangements must be done in order to ensure crawling performance and corpus update, ultimately improving search experience.

Despite more than 20 years of evolution in web crawling since the WWWW (World Wide Web Wanderer) project [Baeza-Yates and Ribeiro-Neto, 2011], as the Web evolves, new challenges arise. Besides dealing with some practical issues, such as DNS resolution, URL canonization, page redirections, broken links, and download errors, current crawlers must address some difficult problems. In the following, we present some of such problems:

Freshness: One of the goals of every crawler is to keep the local corpus of documents up-to-date. However, the refresh rate of documents varies dramatically in the Web [Fetterly et al., 2003b]. While some documents change over long periods of time, others change multiple times a day. Depending on the interest of users and the quality of such documents, they should be collected as soon as possible, keeping the latest version in the local corpus to improve search experience. With this in mind, a challenging problem is to decide how often should the crawler revisit a particular URL. Moreover, how to quickly discover and download brand new web documents which potentially interest users is also a challenge.

Coverage: According to Lawrence and Giles [1998], the coverage of the search engines is limited. No single search engine indexes more than about one-third of the documents on the Web. Although some search engines are only interested in certain

[2]URL is an acronym for uniform resource locator, a reference to a document in the Web.

topics, coverage is an important issue, since it directly impacts the search experience. Therefore, collecting a comprehensive set of documents of the Web is a key problem in web search.

Duplicates: A large portion of web documents have similar content. According to Fetterly et al. [2003a], 29% of the documents on the Web are near duplicates, and 22% of them are identical. Detection of redundant content is an important issue, which facilitates the removal of near duplicates and mirrors, in order to reduce crawling cost, by reducing network traffic, as well as to improve search experience. Thus, another challenging problem is to avoid waste of resources crawling duplicate content.

Web spam: According to Spirin and Han [2012], the amount of web spam varies from 6% to 22%, generating a negative impact on search. Spam deteriorates the quality of search results, weakens trust of users in search engines, which is critical due to the zero cost of switching from one search engine to another. Additionally, it deprives legitimate revenue websites, potential costumers of commercial search engines, of earning in a noise free environment. Moreover, web spam serve as means of malware and phishing attacks.[3] In this scenario, a difficult challenge is to avoid the crawling of web spam.

Deep Web: Different from the surface Web, where documents are reachable by following hyperlinks between neighbors, in the deep Web, the documents are not easily reachable, since they are generated dynamically, typically in response to a user action, such as form submissions or entering a protected area [Madhavan et al., 2008]. As a result, the deep Web is hundreds of times larger than the surface Web [Chang et al., 2004]. With this in mind, crawling hidden documents in the deep Web is a challenging problem.

2.1.2 Indexing

The indexing process consists of building data structures to efficiently represent the content in the local corpus in order to speed up the searches [Witten et al., 1999]. In search engines, the indexing process is typically performed by the indexer component.

As shown in Figure 2.1, the indexer parses documents from the local corpus, extracting textual elements, such as sentences, and phrases, for later transformations. Such transformations are then performed by the *tokenizer* and involve breaking down

[3]Malwares are malicious softwares used to disrupt computer operation, or gain access to private computer systems, while phishing is the attempt to gather sensitive information, such as usernames, passwords, and credit card details.

the textual elements into tokens. Finally, the *analyzer* performs multiple text operations on individual tokens, such as stopwords[4] removal, stemming[5], and token categorization, recording their occurrences in each document. As a result, two data structures are created: the *vocabulary*, which records the set of all selected tokens, and the *occurrences*, which records, for each token in the vocabulary, the documents in the local corpus which contain the token. Optionally, the relative positions of the token into the documents are also recorded [Baeza-Yates and Ribeiro-Neto, 2011]. Together, these two data structures comprise the inverted index, the core of modern indexing architectures [Witten et al., 1999], and by far the preferred choice to implement information retrieval systems.

Besides dealing with non trivial efficiency issues, current indexers must also address other difficult and challenging problems. In the following, we present an excerpt of such problems:

Large corpora: Based on the number of uniquely addressable documents reported by web search engines, the amount of documents in the surface Web increased 1,500 times over the last 8 years, raising from 20 billion to 30 trillion [Gulli and Signorini, 2005; Bar-Yossef and Gurevich, 2008; Cutts, 2012]. Considering the deep Web, the growth is undoubtedly higher. Moreover, there is no evidence that this growth will stop or even slow down. With this in mind, even very simple structures, such as inverted indexes, require sophisticated implementations in order to achieve reasonable performance on very large corpora [Baeza-Yates and Ribeiro-Neto, 2011]. For instance, operating an index in compressed form was not recommended until recently, but the increasing size of indexes, and the performance gap between processors and external devices changed the way of thinking. Additionally, the possibility of implementing complex but effective retrieval models over large collections lead to new trade offs not previously considered.

Vocabulary size: The natural consequence of a larger corpus, is a large index. So, a difficult problem is to keep vocabulary in control, since in an inverted index the retrieval efficiency drastically drop as the vocabulary increases. Despite the possibility to apply operations that reduce the size of the vocabulary, the syntax of some languages poses difficult issues to search engines. Particularly, typical indexing of agglutinating languages, such as German, or Eastern languages, such as Japanese and Chinese that represent texts as sequences over a very large alphabet of ideograms, result in very

[4]Stopwords are tokens with a high frequency in documents, but with little discriminative power of such documents.

[5]A comprehensive explanation on stemming is provided in Section 2.2.

large vocabularies [Manning et al., 2008; Baeza-Yates and Ribeiro-Neto, 2011]. Thus, the challenge is to control the vocabulary size while handling different languages.

Noise and multiple text formats: The democratic and self-regulating nature of the Web allows users to publish content without correctness and accuracy restrictions. Accordingly, there is no guarantees on the quality of the published content. Beyond the unintended noise, such as misspellings and poor formatting, web documents typically comprise irrelevant content besides their core topic, such as advertisements and scripting code, not to mention web spam. In addition, web documents present a variety of content types and character encodings [Croft et al., 2009]. In this environment, effectively parsing web documents is a complex and challenging task.

Compression: The motivation for compressing text is that consuming less storage space and less bandwidth for transmission means spending less money. However, the price paid is that some computing time is required to encode and decode text [Witten et al., 1999]. The problem of efficiently representing information is nothing new, and despite increase in storage and transmission capacities, more effort has been put into using compression to increase the amount of data that can be handled. The fast growth of the Web contributed to this problem by posing new scaling issues. With this in mind, the efficient compression of inverted indexes becomes paramount to provide fast retrieval performance with low storage cost.

2.1.3 Querying

Querying is the process of answering queries posed by users in information retrieval systems. The users specify a set of words that convey the semantics of their information need, and wait for the system to deliver relevant answers that meet their need [Baeza-Yates and Ribeiro-Neto, 2011]. This waiting time is known as query latency. In search engines, the querying process is performed by the query processor component.

Figure 2.1 illustrates the querying process flow. It starts with the *logging* procedure receiving and storing the query posed by the user into the query log repository. Next, the *understanding* procedure identifies the underlying user information need, by performing operations such as spelling correction, acronym expansion, stemming, query reduction, query segmentation, and query classification, ultimately reformulating the original query in order to improve retrieval effectiveness. A particularly effective query understanding operation that can be also performed is query expansion, which involves

adding terms from the local corpus or from external knowledge bases to the original query, optionally reweighting the original or added terms, and generating the final system query. Lastly, the *ranking* procedure employs a retrieval model to match the system query and the inverted index in order to deliver a ranking of relevant documents, which properly meet the user's information need.

Ranking is a crucial task in search engines, since the retrieval effectiveness depends ultimately on the quality of the delivered ranking. By quality we mean bringing all the relevant documents on the corpus which fulfil the user information needs, while avoiding non-relevant documents as much as possible. Despite the large number of documents typically retrieved for a query, the user is normally only willing to inspect for relevance in the very few top ranking positions [Silverstein et al., 1999]. In fact, more than three quarters of users inspects only the top 5 positions [Chikita, 2013]. In order to improve the web search ranking, Santos [2013] presents a comprehensive description of the recent research on this topic, which shows that ranking in search engines is definitely a challenging task.

A key requirement for delivering high quality rankings is effectively interpreting the search intent underlying the user's query. Due to the particular interest on query understanding for this thesis, the next section cover this topic by describing typical query understanding operations, and different approaches in the literature to address the query expansion problem.

2.2 Query Understanding

The query representation and understanding research field have received special attention by the information retrieval community. Recently, two workshops organized by the ACM SIGIR (Special Interest Group on Information Retrieval) were dedicated to bring together the different strands of research on this topic [Croft et al., 2011]. They aim to develop common themes and directions in query understanding research, including definitions of tasks, evaluation methodology, and reusable data collections.

According to Li [2010], query understanding aims to derive a representation of the user's query better suited for a search engine. An improved understanding of the information need underlying the user's query becomes paramount for an improved search experience. Indeed, misinterpreting this query may result in relevant documents never being retrieved, regardless of how sophisticated the subsequent ranking process is. In practice, query understanding improves the initial query formulation by applying different operations on the user queries, paying the cost of a negative impact on the

query latency. In the following, we provide an excerpt of typical query understanding operations.

Stemming: The process of transforming inflected or derived terms to their root form is known as stemming [Porter, 1980]. This process allows a query term to match documents containing all forms of the term. For instance, the term "walk" will match "walking", "walked", and "walks". Stemming can be done either on the terms in a document during indexing and on the query during querying, or by expanding the query with the variants of the terms during querying. Although traditional stemming increases recall by matching term variants, it potentially reduces precision by retrieving too many documents that have been incorrectly matched. According to Peng et al. [2007], one needs to be very cautious when using stemming in web search engines, since its blind transformation of all query terms, always performing the same transformation for the same term without considering the query context, and its blind matching of all occurrences in documents can negatively impact the retrieval effectiveness.

Spelling correction: The spelling correction process consists in transforming misspelled terms to the correct form. According to Cucerzan and Brill [2004], more than 10% of queries sent to search engines contain misspelled terms, reducing the chances of document matching for such queries. This statistic suggests that an effective query speller is crucial to improve the retrieval performance in web search engines. However, as observed by the authors, general purpose spelling correction methods commonly perform poorly in web queries, demanding the development of more sophisticated approaches for this environment. Particularly, the learning approach proposed by Li et al. [2006] outperforms general purpose query spelling approaches in the literature, becoming a suitable alternative for spelling correction in web search.

Acronym expansion: Acronyms are abbreviations or short descriptors of phrases, formed from the initial letters of the important terms in a phrase. For example, "UN" is an abbreviation for the "United Nations", and "UNICEF" is used as an abbreviation for the "United Nations Children's Fund". Acronym usage is becoming more common in web searches, email, text messages, tweets, blogs and posts [Taneva et al., 2013]. The process of transforming an acronym to their original phrases is known as acronym expansion [Jain et al., 2007]. Traditional approaches rely on the presence of text markers or linguistic cues, assuming that acronyms are introduced in a more formal manner [Park and Byrd, 2001; Nadeau and Turney, 2005]. However, on the Web, acronym-expansion pairs often do not occur in the same sentence, nor do they follow the

familiar pattern of being formed from its full form's leading characters [Zhang et al., 2011]. Furthermore, acronyms are typically ambiguous and often disambiguated by context terms. In a recent work, Taneva et al. [2013] showed that supervised learning approaches are the most effective alternative for acronym expansion on the Web.

Query segmentation: The process of separating query terms into segments so that each segment maps to a semantic component, or concept, is called query segmentation [Risvik et al., 2003]. For instance, for the query "obama family tree", segmenting it in the concepts "obama" and "family tree" is good, while segment it in the concepts "obama", "family", and "tree" is not, since the "tree" segment greatly deviates from the intended meaning of the query. Query segmentation commonly improves retrieval effectiveness, since segments carry implicit term proximity and ordering constraints, which may be used to efficiently match documents and queries. Different approaches have been proposed to address this problem. While unsupervised approaches use well known features and statistical modeling to capture correlations among terms based on query log data and knowledge bases [Risvik et al., 2003; Jones et al., 2006; Tan and Peng, 2008], supervised approaches train a machine learning mechanism using multiple features to generate a model which can effectively predict correlation among terms [Bergsma and Wang, 2007].

Query classification: The query classification process consists in associating classes from a list of predefined target categories to user queries. Such category information can be used to trigger the most appropriate vertical searches to a query, improving document ranking [Cao et al., 2009]. Classifying queries is a more challenging task than classifying text, due to the dynamic nature of web content, users and query traffic [Beitzel et al., 2005], the very short size of web queries, which contains from 2.4 to 2.7 terms on average [Jansen et al., 2000; Gabrilovich et al., 2009], and query ambiguity [Cui et al., 2002], which typically leads to a query belonging to multiple categories. An effective approach for this task uses the feedback information provided by the own search results as a source of knowledge for classification [Gabrilovich et al., 2009]. The assumption of these approaches is that classifying the highest ranking documents for a query provides an indirect way to classify such query. Additionally, Cao et al. [2009] show that context-aware approaches are more effective for query classification. Their proposed approach leverages context information to classify queries by modeling search context through conditional random fields (CRF), a framework of probabilistic models for segmenting and labeling sequence data [Lafferty et al., 2001].

Query reduction: According to Kumaran and Allan [2008], richer expressions of information need, taking the form of longer than usual queries, can be leveraged to improve search performance. However, handling long queries is difficult as they usually contain a lot of noise, which means extraneous terms that the user believes are important to convey the information need, but in fact are confusing to search engines. The process of pruning a long query to retain only important terms, ultimately creating a more concise query, is called query reduction. Although rare in the Web, long queries are common in some specific applications [Ganguly et al., 2011]. Furthermore, web queries have grown in size at a yearly rate of 10%, while single term queries have dropped 3%, demanding the development of sophisticated query reduction techniques suitable for the Web [Kumaran and Carvalho, 2009; Balasubramanian et al., 2010].

In the remainder of this section, we describe other effective query understanding operations. Particularly, Section 2.2.1 describes two effective feedback methods for query reformulation. Section 2.2.2 extends the discussion by focusing on one of such feedback methods, also presenting traditional and novel approaches to handle query expansion, and Section 2.2.3 further extends the discussion by presenting the main query expansion approaches reported in the literature.

2.2.1 Feedback Methods

In addition to the aforementioned query understanding operations, there are two methods particularly interesting to promote query reformulations: *relevance feedback* and *query expansion*. Figure 2.2 illustrates the basic operation of these methods.

While in relevance feedback the users explicitly provide information on relevant documents to a query, in query expansion, information related to the query is used to expand it [Baeza-Yates and Ribeiro-Neto, 2011]. Both methods reformulate the original query q by adding terms to it, optionally reweighting original and added terms, finally generating a new system query q'. However, they differ in the way they determine feedback information that is either related or expected to be related to the original query. In relevance feedback, the feedback is explicit with users or a group of human assessors directly providing feedback information by inspecting the top ranked documents for the original query formulation, and indicating those indeed relevant to the query. In the Web, a less expensive and time consuming approach can be used without disrupting the users. The user's clicks on search results constitute an alternative way to indirectly capture relevance information for a query. Differently, in query expansion, the feedback is commonly implicit, with feedback information being derived by the system without the participation of the user.

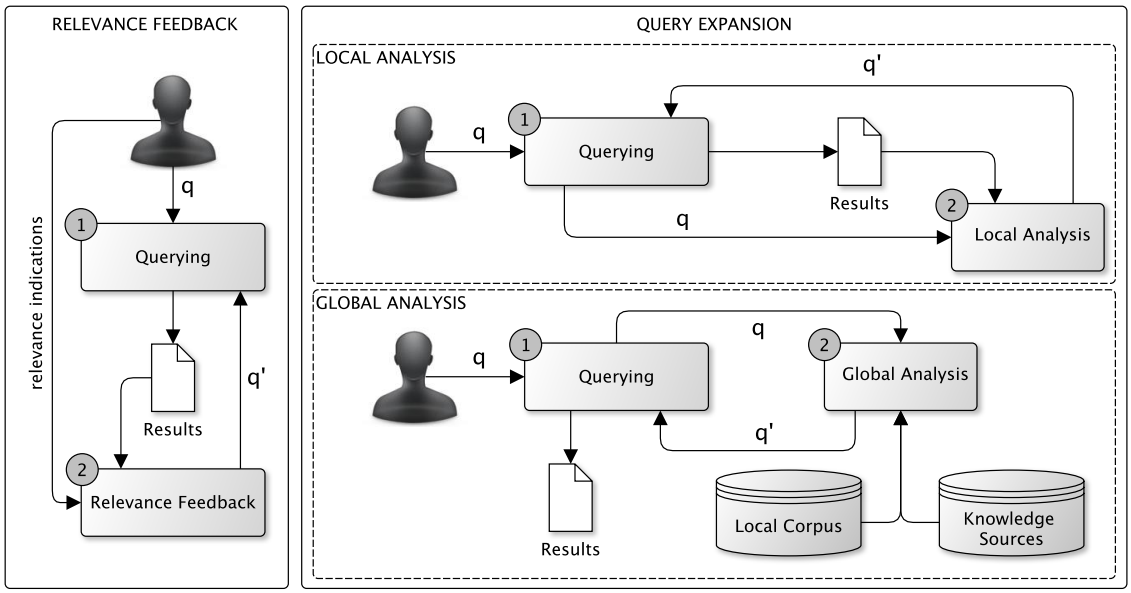

Figure 2.2. The relevance feedback and query expansion methods.

Particularly, there are two basic techniques for compiling implicit feedback information in query expansion: *local analysis*, where feedback information is derived from the top ranked documents in the result set for the original query formulation, and *global analysis*, where feedback information is derived from the local corpus or from external knowledge sources. Typically, query latency is greater when using local analysis than global analysis, since a preliminary ranking must be processed in order to provide the top k documents from where the expansion terms are extracted. Additionally, as in global analysis the feedback information is not necessarily related to the original query, its utilization is more challenging than local analysis and explicit feedback. Furthermore, the compilation of feedback information in large scale is one of the main challenges in query expansion [Baeza-Yates and Ribeiro-Neto, 2011].

In the next section, we extend the discussion on query expansion by presenting approaches that use information on entities to improve the selection of candidate expansion terms.

2.2.2 Query Expansion

Query expansion is an effective query understanding operation, which aims to enhance the representation of the user's initial query by appending useful terms to it, optionally reweighting original and appended terms, in an attempt to improve retrieval performance [Lavrenko and Croft, 2001; Zhai and Lafferty, 2001b]. It is motivated by

the fact that users are frequently unable to choose the best terms to formulate their queries, often using only a couple of terms [Jansen et al., 2000; Gabrilovich et al., 2009] when searching the Web, which may lead to poor results.

As mentioned in Section 2.2.1, query expansion is a feedback method that compiles feedback information through local or global analysis. In the following, we present the typical query expansion process that uses the local analysis technique, also known as *pseudo-relevance feedback*, as well as the recently proposed entity-oriented query expansion process that uses global analysis to perform the expansion.

Pseudo-Relevance Feedback

The pseudo-relevance feedback (PRF) process consists in extracting expansion terms from a set of feedback documents, appending them to the user query in order to close the gap between the vocabulary of the query and that of potentially relevant documents. The feedback set comprises the top retrieved documents for the query, which are assumed to be relevant [Rocchio, 1971; Lavrenko and Croft, 2001; Zhai and Lafferty, 2001b]. Figure 2.3 illustrates the pseudo-relevance feedback process.

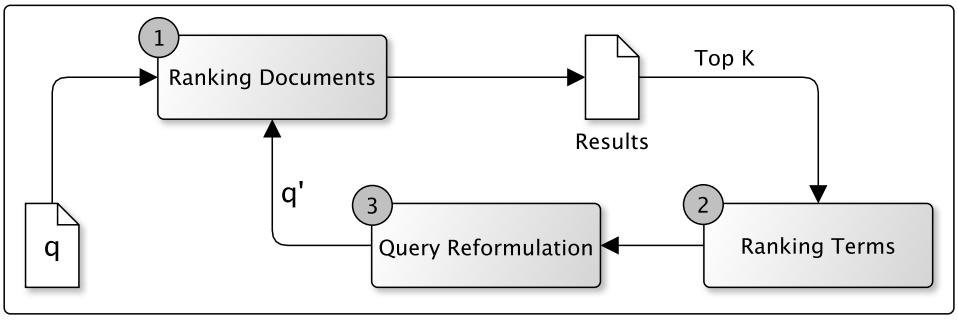

Figure 2.3. The pseudo-relevance feedback process.

From the initial user query q, the search engine ranking procedure produces an initial result set, a ranking of potentially relevant documents (step 1 in Figure 2.3). Next, from the top k best ranked documents on the initial result set, the candidate expansion terms are extracted and ranked (step 2 in Figure 2.3). In order to rank such candidate expansion terms, various techniques can be used, such as local clustering [Attar and Fraenkel, 1977] and local context analysis [Xu and Croft, 1996]. In the first case, terms are clustered based on their frequency of co-occurrence within the top k best ranked documents, and a score is computed for each term considering its distance from query terms within the cluster more related to the query. In the second case, groups of terms are derived from the top k best ranked documents and scored based on

their similarities to the whole query q using a variant of TF-IDF ranking. Next, the top m terms, or groups of terms depending on the ranking technique adopted, are added to the original query q generating a new system query q' (step 3 in Figure 2.3). Lastly, the search engine ranking procedure produces a final result set, ultimately delivering it to the user.

Despite widely used in modern search engines [Baeza-Yates and Ribeiro-Neto, 2011], pseudo-relevance feedback is based on the often invalid assumption that the top retrieved documents are relevant to the user query, which may lead to an improper expansion that may cause the subsequent ranking process to drift away from the user's information need [Mitra et al., 1998]. In order to overcome this limitation, improved mechanisms for selecting effective feedback documents [He and Ounis, 2009], and effective expansion terms [Cao et al., 2008; Lee et al., 2009; Udupa et al., 2009] have been proposed. Additionally, recent entity-oriented pseudo-relevance feedback (ePRF) approaches that promote in the initial result set the documents which refer to named entities cited in the query have also been shown to be effective for query expansion [Xu et al., 2009].

Entity-Oriented Query Expansion

Recently, entity-oriented query expansion approaches that exploit knowledge bases as repositories of high-quality feedback *entities* have been proposed, mostly based on the observation that queries with named entities, such as persons, organizations, and locations, account for over 70% of the web search traffic [Guo et al., 2009]. Typically, such approaches rely on the availability of structured information about named entities identified in the user's query to derive effective expansion terms. The entity-oriented query expansion process underlying such approaches consists of extracting candidate terms from an entity identified in the query, appending the top m best candidates to the user query. Additionally, the original and appended terms can be reweighted in order to improve expansion performance. Figure 2.4 illustrates the entity-oriented query expansion process.

In the off-line stage of the process, an entity repository is created by extracting entity-related information from the Web (step 1 in Figure 2.4). In the on-line stage, given a user query q, a named entity recognition task is triggered in order to identify an entity in q (step 2 in Figure 2.4). If no such entity is found, q is not expanded and the search engine ranking procedure produces the results considering q (step 5 in Figure 2.4). Otherwise, candidate terms related to the recognized entity, as recorded in the entity repository, are ranked given the query q using a ranking procedure (step 3

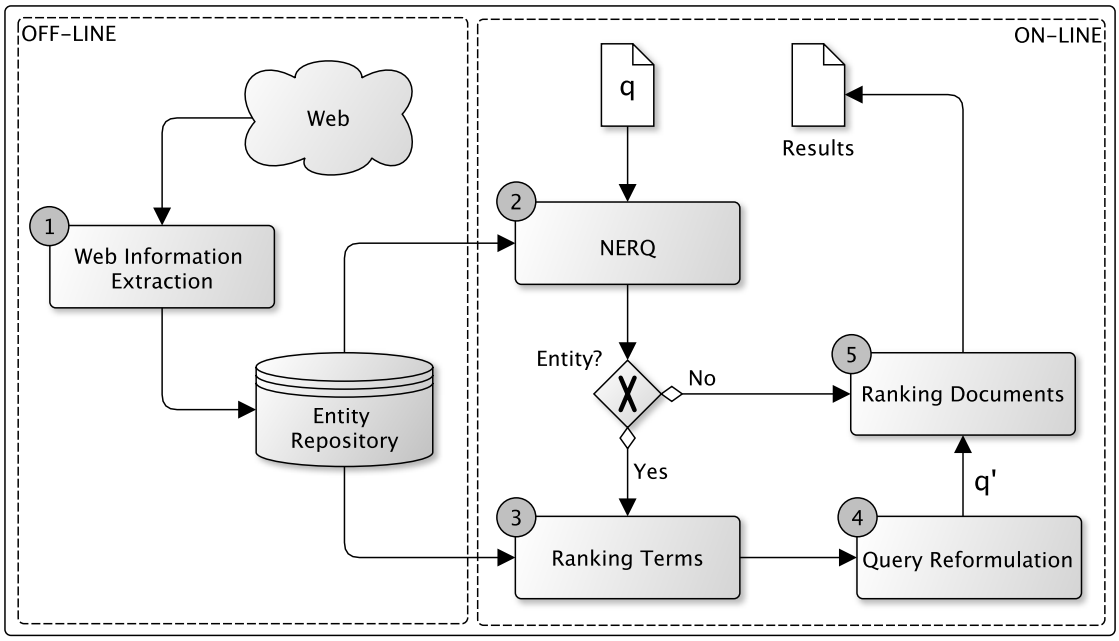

Figure 2.4. The entity-oriented query expansion process.

in Figure 2.4). Lastly, the top m best ranked terms according to the ranking procedure are appended to q in order to produce the expanded query q' (step 4 in Figure 2.4), which will be then used by the search engine to retrieve the final ranking of results to be presented to the user (step 5 in Figure 2.4).

Particularly, steps 1 and 2 in Figure 2.4 refer to well known problems in the literature, which are information extraction (IE) and named entity recognition (NER), respectively. In the following, we contextualize them.

Information Extraction: According to Chang et al. [2006], information extraction is the task of automatically translating an input into a target, where the input can be unstructured documents, such as free text written in natural language, or semi-structured documents that are pervasive on the Web, such as tables or itemized and enumerated lists, and the target can be a relation of k-tuple, with k being the number of attributes in a record, or a complex object with hierarchically organized data. Typically, such a task is performed by an automatic extraction system referred to as extractor or wrapper. Different from information retrieval, which concerns how to identify relevant documents from a document collection, information extraction produces structured data ready for post-processing, which is crucial to many web mining and web search applications [Hu et al., 2011].

While traditional information extraction approaches aim at extracting data from

totally unstructured free texts that are written in natural language, web information extraction (Web IE) approaches process on-line documents that are semi-structured and usually generated automatically by a server-side application. As a result, traditional information extraction approaches usually take advantage of natural language processing (NLP) techniques, such as lexicons and grammars, whereas Web information extraction approaches usually apply machine learning and pattern mining techniques to exploit the syntactical patterns or layout structures of the template-based documents [Habegger and Quafafou, 2004; Yun and Seo, 2006; Gatterbauer et al., 2007; Hu et al., 2011].

Named Entity Recognition: Early work in information extraction was mostly inspired by the Message Understanding Conferences [Grishman and Sundheim, 1996], which defined five main tasks for textual information extraction, including named entity recognition. Named entity recognition involves processing text to identify important content-carrying units, the named entities, and classify them into predefined categories, such as persons, organizations, locations, biological species, temporal expressions, books, music, quantities, monetary values, and percentages [Mikheev et al., 1999; Whitelaw et al., 2008]. Since the early 1990s, the interest in this information extraction task has been growing, particularly because named entity recognition plays a vital role in several applications, specially question answering systems, and web search.

Typically, named entity recognizers perform the identification and classification of entities using a statistical model that learns patterns from manually-annotated textual corpora [Mikheev et al., 1999; Chieu and Ng, 2003]. Despite the near-human performance of such traditional recognizers, they are particularly ineffective to recognize named entities in the absence of annotated data, demanding mechanisms that provide such annotations [Mika et al., 2008; Richman and Schone, 2008] and create an automatically-annotated textual corpora competitive to existing gold-standards [Nothman et al., 2013].

Recently, named entity recognizers that use efficient machine learning algorithms and annotated textual corpora have been shown to be effective for named entity recognition [Zhou and Su, 2005; Nadeau et al., 2006], particularly those that use conditional random fields extractors for this task [McCallum and Li, 2003; Liao and Veeramachaneni, 2009]. However, considering the Web, one needs effective approaches to automatically generate web-scale training data and to perform on-line classification, recognizing not only high level categories, such as places and persons, but also more fine-grained categories, such as soccer players, protein names, and universities [Whitelaw et al., 2008].

Considering web search, a particularly interesting sub-task of named entity recognition is the detection of a named entity in a given query, a problem known as named entity recognition in queries (NERQ) [Paşca, 2007; Guo et al., 2009]. Beyond its obvious applicability in the entity-oriented query expansion problem, it is potentially useful in many other web search applications, such as vertical search engines, advertisement and product recommendation.

2.2.3 Related Approaches

Throughout this chapter, we contextualized web search in general, and query understating operations in particular, with a further look into the query expansion task. In this section, we review query expansion approaches from the literature focusing on the ones which outperform pseudo-relevance feedback, particularly improved approaches for selecting effective expansion terms, effective feedback documents, and effective feedback entities.

Selection of expansion terms: Regarding an improved selection of expansion terms, Cao et al. [2008] found that a non-negligible fraction of expansion terms identified by traditional pseudo-relevance feedback approaches is either neutral or harmful to the effectiveness of the initial query. As a result, they proposed a supervised classification approach using support vector machines (SVM) to predict the usefulness of expansion terms. In a similar vein, Udupa et al. [2009] found that the usefulness of a term may vary drastically depending on the already selected terms. Hence, they proposed to take into account term interactions in order to identify a useful set of expansion terms. Their approach was based on a spectral partitioning of the weighted term-document matrix using singular value decomposition (SVD). Both approaches showed significant improvements compared to state-of-the-art pseudo-relevance feedback approaches, such as relevance models [Lavrenko and Croft, 2001] and model-based feedback [Zhai and Lafferty, 2001b]. Focusing on difficult queries, Kotov and Zhai [2012] conducted a study on methods leveraging the ConceptNet knowledge base to improve the search results for these poorly performing queries. They proposed a supervised approach using generalized linear regression to use concepts from ConcetpNet to expand difficult queries.

Selection of feedback documents: Regarding the selection of feedback documents, several approaches have been proposed to leverage high-quality external resources as feedback. Particularly, Cui et al. [2002] used the log of the queries submitted by pre-

vious users to establish probabilistic correlations between query terms and document terms, narrowing the gap between the query space and the document space. In addition, He and Ounis [2007] found that the performance of query expansion can be improved by using a large external collection. As a result, they proposed an adaptive query expansion mechanism based on Wikipedia that predicts the query expansion performance and decides if the expansion should be done and which collection (internal or external) should be used for the expansion.

In a similar vein, Milne et al. [2007] found that an external knowledge source, such as Wikipedia, with a vast domain-independent pool of manually defined terms, concepts and relations, is useful for query expansion. Hence, they proposed a mechanism to automatically derive a thesaurus from Wikipedia and used such thesaurus to expand queries, ultimately allowing users to express their information needs more easily and consistently. Still on Wikipedia, Li et al. [2007] observed that the content quality and the evolving nature of this external knowledge base make it an important resource for query expansion. As a result, they exploit Wikipedia as an external corpus to expand difficult queries. Furthermore, they emphasized that some features which could be extracted from it, such as, links, sections and references, remain unexplored to accurately select effective expansion terms.

In a different vein, Lin et al. [2011] found that social annotation collections can be seen as a manually edited thesaurus which provide more effective expansion terms than pseudo-feedback documents. Hence, they proposed a machine learning mechanism that extracts candidate expansion terms from a social annotation collection using a term-dependency method, later ranking such terms in order to select the top k best ranked for expansion. The learning approach showed significant improvements compared to state-of-the-art pseudo-relevance feedback approaches, such as relevance models [Lavrenko and Croft, 2001] and model-based feedback [Zhai and Lafferty, 2001b]. Recently, Bendersky et al. [2012] leveraged multiple sources of information, such as a large collection of web n-gram counts, anchor and heading text extracted from a large web corpus, articles and titles from Wikipedia, search engine query logs, and the documents in the retrieval corpus, for selecting a relevant and diverse set of expansion terms. In the same vein, Weerkamp et al. [2012] proposed a general generative query expansion model that uses a mixture of external collections, such as news, blog posts, Wikipedia, and web documents for query expansion.

Selection of feedback entities: Another effective approach to query expansion exploits knowledge bases as repositories of feedback *entities*—as opposed to feedback documents. Such an entity-oriented pseudo-relevance feedback approach relies on the

availability of structured information about named entities identified in the user's initial query [Guo et al., 2009]. For instance, [Xu et al., 2008, 2009] proposed an entity-oriented pseudo-relevance feedback approach that recognizes the most representative entity in a query, and uses Wikipedia articles related to this entity as feedback documents for query expansion. In their approach, the top terms extracted from the feedback documents are selected according to the terms' likelihood of being effective, as predicted by an SVM classifier, and appended to the initial query. This approach was shown to outperform a state-of-the-art pseudo-relevance feedback approach based upon relevance models [Lavrenko and Croft, 2001]. In a different vein, Oliveira et al. [2012] proposed an unsupervised approach based on tag recommendation for query expansion, which considers heuristic metrics extracted from Wikipedia articles to estimate the descriptive capacity of candidate expansion terms, ultimately weighting and ranking those terms in order to use them to improve the retrieval performance. Experiments attested the effectiveness of their approach, showing significant gains on different evaluation metrics compared to a state-of-the-art entity-oriented pseudo-relevance feedback approcach [Xu et al., 2009].

2.3 Summary

This chapter provided a comprehensive and up-to-date background on web search in general, and on query expansion in particular. Starting with an overview of the typical operation of a web search engine, in Section 2.1, we described the processes of crawling, indexing, and querying. Within the scope of the latter, in Section 2.2, we provided a contextualized background on query representation and understanding. This encompassed classical query understating operations, including stemming, spelling correction, acronym expansion, query segmentation, query classification, and query reduction. Additionally, in Section 2.2.1, we described the relevance feedback and query expansion methods to perform query reformulation. Moreover, in Section 2.2.2, we extended the discussion on query expansion, presenting the pseudo-relevance feedback and the entity-oriented query expansion processes. We also contextualised the information extraction and the named entity recognition problems often underlying the entity-oriented query expansion process. Lastly, in Section 2.2.3, we reviewed different approaches for query expansion reported in the literature, with a further look into the most recent ones which consider entity-related information to select expansion terms.

In the next chapter, we introduce a novel unsupervised entity-oriented query expansion approach, which exploits the semantic structure provided by Wikipedia in-

foboxes, and adapt well-known discriminative features to deal properly with entities, in order to select effective expansion terms.

Chapter 3

Unsupervised Entity-Oriented Query Expansion

Query expansion approaches typically append additional terms extracted from the top retrieved documents for the query, which are assumed to be relevant [Lavrenko and Croft, 2001; Rocchio, 1971; Zhai and Lafferty, 2001b]. This assumption is often invalid and may cause the expansion process to drift away from the user's information need [Mitra et al., 1998].

Recently, effective entity-oriented query expansion approaches have been proposed in the literature [He and Ounis, 2007; Li et al., 2007; Milne et al., 2007; Xu et al., 2008, 2009; Oliveira et al., 2012]. Particularly, they exploit knowledge bases as repositories of feedback *entities*, using them to recognize named entities in queries, and map the recognized entities to ones in repositories in order to obtain effective expansion terms, ultimately reformulating the original queries using the best ranked candidates. However, none of them have properly exploited valuable human-refined information available in the knowledge bases, such as Wikipedia infoboxes, to obtain effective candidate expansion terms.

Differently from previous approaches in the literature, we propose an unsupervised approach that takes advantage of the semantic structure implicitly provided by infoboxes templates, and leverage well-known discriminative features, adapting them to deal properly with entities, ultimately improving their accuracy in selecting effective expansion terms.

The remainder of this chapter describes our unsupervised entity-oriented query expansion approach. In particular, Section 3.1 introduces our approach, describing its retrieval process and main components. Section 3.2 starts by discussing how an entity repository can be derived from Wikipedia, before presenting the entity-related

information actually used by our approach. Section 3.3 describes the entity resolution procedure used to recognize named entities in queries, and map them to their corresponding entry in the entity repository. Section 3.4 describes the ranking procedure, the features and the ranking combination strategies used to rank effective expansion terms. Lastly, Section 3.5 presents the experimental methodology and evaluation procedures used to validate our approach in contrast to a standard retrieval baseline.

3.1 Overview

The presence of a named entity in a query provides an opportunity for web search engines to exploit the rich evidence about this entity available from a knowledge base in order to improve their understanding of the user's information need. In this section, we introduce a query understanding approach that builds upon this idea. Particularly, we propose an unsupervised entity-oriented query expansion approach called UQEE, an acronym for "Unsupervised Query Expansion using Entities". The retrieval flow of UQEE is illustrated in Figure 3.1.

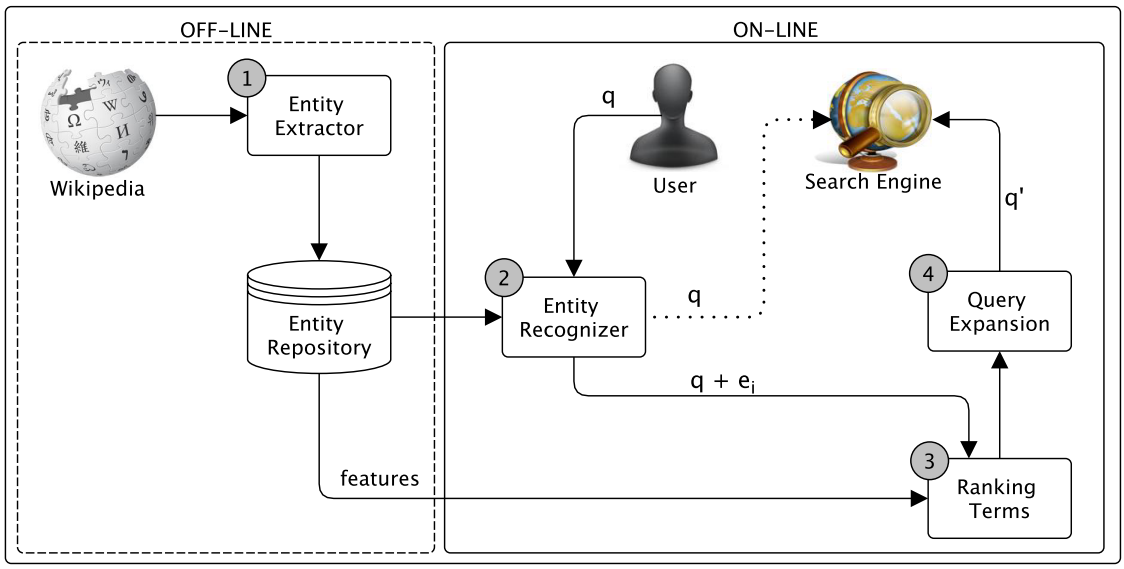

Figure 3.1. The off-line and on-line query processing with UQEE.

The off-line stage of UQEE is responsible for assembling an entity repository W by processing a knowledge base (step 1 in Figure 3.1). In the on-line stage, given a user query q composed of a sequence of l terms $\{t_1\ t_2\ \cdots\ t_l\}$, UQEE generates a new expanded query q'. To this end, it attempts to recognize in the query q a named entity e_i from the repository W (step 2 in Figure 3.1). If no such entity is found, q is

not expanded. Otherwise, candidate terms related to e_i, as recorded in the repository W, are ranked given the query q using a ranking procedure (step 3). Lastly, the top k ranked terms according to the ranking procedure are appended to q in order to produce the expanded query q' (step 4), which will be then used by the search engine to retrieve the final ranking of results to be presented to the user.

Most of the work of UQEE is done at the off-line stage, and the computational cost of the on-line stage is negligible. Similarly to standard pseudo-relevance feedback approaches, the query latency of our approach is primarily affected by the number of terms considered for expansion. However, standard pseudo-relevance feedback approaches need to process both the original query, in order to select terms for expansion, and the modified one, while UQEE only processes the modified query. Thus, the computational overhead of UQEE in query time is lower than that of standard pseudo-relevance feedback approaches.

3.2 Entity Representation

Our unsupervised entity-oriented query expansion approach builds an entity repository W using Wikipedia, a free on-line encyclopedia which enables collaborative publication and dissemination of ideas and concepts. Due to its popularity, coverage, accessibility, multilingual support, and high quality content, Wikipedia has rapidly turned into an important lexical semantic resource on the Web. Indeed, it has shown a strong potential to attenuate knowledge acquisition bottlenecks and coverage problems found in current lexical semantic resources [Zesch et al., 2007].

Wikipedia comprises semi-structured documents, known as articles, with each article describing a named entity, such as a person, organization, or location. The textual content of Wikipedia is available for download within periodically released database dumps[1]. We designed a special-purpose parser that processes Wikipedia articles in order to populate W. This process includes lower case conversion, stopword removal, and stemming using Porter's stemmer [Porter, 1980]. In addition, during indexing, we discard numeric terms, non-English terms, and terms with special characters.

Consider an entity repository W comprising a set of p entities $E = \{e_1, e_2, \cdots, e_p\}$. Each entity $e_i \in E$ is represented as a tuple $e_i = \langle F_i, A_i, c_i \rangle$, where F_i and A_i are the sets of fields and aliases of e_i, respectively, and c_i is the class to which this entity belongs. In particular, each field in F_i comprises textual content from a

[1] `http://dumps.wikimedia.org/`

specific region in the article that describes the entity e_i. Table 3.1 presents the article fields considered by our unsupervised entity-oriented query expansion approach.

Table 3.1. The article fields considered by UQEE.

Field	Description
infobox relationship	anchor text of hyperlinks in the infobox
infobox property	textual content (no anchor text) in the infobox
content	full textual content of the article

As an illustrative example, Figure 3.2 shows an excerpt of the Wikipedia article describing the entity *Barack Obama*, highlighting the fields *title*, *summary*, *category*, and *infobox*.

Figure 3.2. The Wikipedia article describing the entity *Barack Obama*.

An entity in Wikipedia may be referred to by multiple names. For instance, as illustrated in Figure 3.2, in Wikipedia, the names "Obama", "Barack Obama", and "44th President of the United States"—to name a few—are all alternative entry points for the single article representing the entity *Barack Obama*. In order to improve the recognition of named entities in web search queries, we use these multiple names as the set of aliases A_i of the entity e_i.

Lastly, entities in Wikipedia can be classified in different manners. For instance, Wikipedia contributors may assign each article a category, such as those described at the bottom of Figure 3.2 for the article describing the entity *Barack Obama*. While these categories are leveraged by our query expansion approach as a textual field, they are less suitable for assigning each entity a unique class. As we will see in Section 3.4.1, such a unique class is useful for identifying informative terms, i.e., terms that are useful descriptors of a particular entity as opposed to those that describe several entities of the

same class. To this end, we exploit infobox templates as a means to identify the single most representative class of an article. In particular, infobox templates are pre-defined sets of attributes that can be used to build an infobox. For instance, in Figure 3.2, the pre-defined attributes *Vice president* and *Preceded by* of the infobox template *president* are used to build the infobox of the entity *Barack Obama*. Accordingly, we choose "president" as the unique class c_i for this entity.

3.3 Entity Resolution

At querying time, we must be able to map the occurrence of a named entity in the query to the corresponding entity in the repository W. However, as discussed in Section 3.2, an entity may be represented by multiple names. Conversely, a single name can be ambiguous, in which case it can refer to multiple entities. As an example of the latter case, consider the entities *Barack Obama* and *Obama, Fukui* (a city in Japan), which can be both referred to by the string "obama" in a query. To overcome these problems, we introduce an entity resolution step ahead of the query expansion.

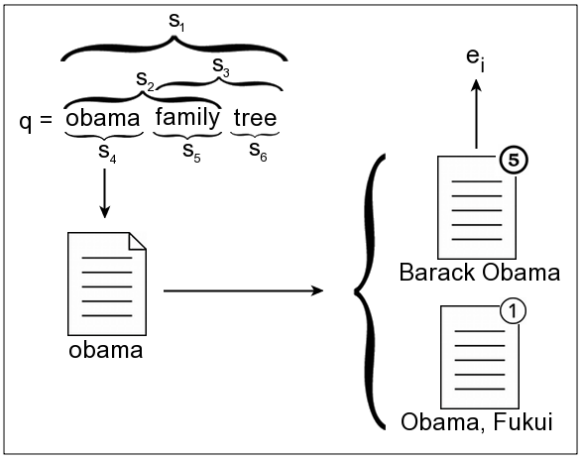

Figure 3.3. The resolution of named entities in a query.

Given a query q with length l, we build a set $S_q = \{s_1, s_2, \cdots, s_z\}$ of all substrings of consecutive terms from q that have a length b, for all $1 \leq b \leq l$. For instance, in Figure 3.3, we have six substrings of consecutive terms extracted from the query *"obama family tree"*: s_1 = *"obama family tree"*, s_2 =*"obama family"*, s_3 =*"family tree"*, s_4 =*"obama"*, s_5 =*"family"*, and s_6 = *"tree"*. The query substrings in S_q are then matched against the aliases of all entities in W. If there is no entity $e_i \in E$ such that $|S_q \cap A_i| > 0$, the query is not expanded, as discussed in Section 3.1. If exactly one entity e_i satisfies this condition, this entity is selected and the resolution process is complete.

For instance, the alias *"obama"* for the entity *Barack Obama* exactly matches the string *"obama"* in S_q. Finally, if there are multiple entities whose aliases match a substring in the query, a disambiguation process is triggered. For instance, the string *"obama"* in S_q also matches an alias of the entity *Obama, Fukui*. In this case, we select the entity with the highest indegree in the Wikipedia graph. For instance, in Figure 3.3, the article *"Barack Obama"* is five times more cited than the article *"Obama, Fukui"*, so the entity represented by the article *"Barack Obama"* is selected. This simple mechanism based on popularity is an effective solution for the experiments we conducted. In a real life scenario, the user intent may not be aligned with what is well cited in Wikipedia. In this case, more sophisticated mechanisms could be easily used by our approach.

Since we can identify multiple entities in a single query, we choose the most representative one as the basis for the subsequent query expansion, so as to avoid drifting away from the user's information need. In particular, given a set of entities $\hat{E} \subseteq E$ identified in a query q, we select the most representative entity $e \in \hat{E}$ as the one with the longest title on Wikipedia. In case of a tie, the entity with the highest estimated quality is chosen. Our premise to select the entity with the longest title on Wikipedia is that longer matches tend to be more specific and hence less prone to ambiguity [Pôssas et al., 2005], which could in turn incur topic drift, a classic side-effect of query expansion.

To estimate the quality of Wikipedia articles, the obvious choice would be the own Wikipedia quality estimators derived from a manual revision process. However, this manual process is becoming infeasible and has been recently criticized by the scientific community [Hu et al., 2007]. The large number of articles, the wide range of subject topics, the evolving content in the articles, the varying contributor background, and abuses contribute to this. Thus, we decided to adopt an automatic machine learning approach to determine the quality of the article that describes an entity in Wikipedia, based upon textual features extracted from this article [Dalip et al., 2009][2]. Specifically, we apply a regression method using the learning algorithm ϵ-Support Vector Regression (SVR) [Vapnik, 1995] to find the best combination of textual features to predict the quality value for any Wikipedia article. Then, we use the predicted quality value of articles to break tied entities. As an example, in Figure 3.3, five distinct substrings of the query *"obama family tree"* can be mapped to entities in the repository W: *"obama family"*, *"family tree"*, *"obama"*, *"family"*, and *"tree"*. The first two substrings are tied with the longest title. Between them, *"obama family"* has the greater quality estimator, and is hence selected as the most representative entity in the query.

[2]Note that, the quality of each article is estimated off-line, during the construction of the entity repository W.

3.4 Ranking Entity Terms

In order to rank effective expansion terms related to the most representative entity identified in the user's query, we introduce a three step ranking procedure which: i) relates a score to each candidate term; ii) discards zero score terms, and; iii) ranks the remaining terms in descending order of their scores. In the remainder of this section, we describe the features used to score candidate expansion terms, as well as the strategies to combine rankings used to instantiate the ranking procedure in our experiments.

3.4.1 Ranking Features

To score candidate expansion terms, we used five features adapted from the literature [Baeza-Yates and Ribeiro-Neto, 2011; Carpineto et al., 2001; Croft et al., 2009]: Dice's coefficient (DC), mutual information (MI), inverse document frequency (IDF), Pearson's CHI-squared (CHI2), and Kullback-Lieber divergence (KLD).

In particular, let t be a candidate term extracted from the article representing the entity e_i, identified from the user's query q. The Dice's coefficient (DC) can be defined according to:

$$\mathrm{DC}(t) = 2 \times \frac{|E_t \cap E_i|}{|E_t| + |E_i|}, \tag{3.1}$$

where E_t is the set of entities that contain the term t and E_i is the set of entities that belong to the same class c_i as the entity e_i. This feature measures the similarity between the sets E_t and E_i. Intuitively, the higher this similarity, the more related to entities in c_i is the term t. Similarly, we can define the mutual information (MI) based upon the sets E_t and E_i, according to:

$$\mathrm{MI}(t) = \begin{cases} |E_t \cap E_i| \times \log \frac{|E_t \cap E_i|}{|E_t| \times |E_i|} & \text{if } |E_t \cap E_i| > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

which measures the mutual dependence—i.e., the amount of shared information—between E_t and E_i. As with the Dice's coefficient, the higher this measure, the more t is related to c_i. Still, based upon the sets E_t and E_i, we can define the inverse document frequency (IDF) according to:

$$\mathrm{IDF}(t) = \begin{cases} \log \frac{|E_t|}{|E_t \cap E_i|} & \text{if } |E_t \cap E_i| > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

which measures the discriminative power of a term in a set of documents. The higher this measure, the more t is related to c_i.

For the other two features, consider $P(t) = |E_t|/|E|$ the probability of a given term t in E, and $P(t|c_i) = |E_t \cap E_i|/|E_i|$ the probability of t given c_i, where E is the set of all entities. We can define the Pearson's CHI-squared (CHI2) based upon $P(t)$ and $P(t|c_i)$, according to:

$$\text{CHI2}(t) = \begin{cases} \frac{(P(t|c_i)-P(t))^2}{P(t)} & \text{if } P(t) \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

which measures the relationship between an expected frequency in the general population ($P(t)$) and an observed frequency ($P(t|c_i)$). The higher this measure, the more t is related to c_i. Similarly, we can define the Kullback-Lieber divergence (KLD) based upon $P(t)$ and $P(t|c_i)$, according to:

$$\text{KLD}(t) = \begin{cases} P(t|c_i) \times \log \frac{P(t|c_i)}{P(t)} & \text{if } P(t|c_i) \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

which estimates the difference between two probability mass functions, in our case $P(t|c_i)$ and $P(t)$. As with the Pearson's CHI-squared, the higher this measure, the more t is related to c_i. Kullback-Lieber divergence is also known as *relative entropy* or *information gain*.

3.4.2 Combining Rankings

In order to combine different rankings produced individually by each one of the features described in Section 3.4.1, we propose two strategies: Ranking frequency (RF), and Borda count (BC).

Ranking frequency is a single-winner election method where candidates are ranked based on the number of different rankings in which they occurred. We score each term using its occurrence on each ranking, according to:

$$\text{RF}(t) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } t \in r_i, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where n is the number of different rankings, t is a candidate term and r_i is the i^{th} ranking. Similarly, Borda count is a single-winner election method based on consensus,

where candidates are ranked in order of preference [Saari, 1999]. We score each term using its position on each ranking, according to:

$$\mathrm{BC}(t) = \sum_{i=1}^{n} p_{t,i}, \tag{3.7}$$

where n is the number of different rankings, $p_{t,i}$ is the position of the term t in the i^{th} ranking r_i, so that if t is not in r_i, $p_{t,i} = k+1$, where k is the number of different terms in each ranking.

3.5 Experiments

In order to validate UQEE, we contrast it to a standard retrieval model across a representative web test collection. In particular, we aim to answer the following research questions:

1. How effective are Wikipedia infoboxes for query expansion?
2. Which features are effective for query expansion?
3. How effective is our unsupervised approach for query expansion?
4. Are automatically generated infoboxes suitable for query expansion?

3.5.1 Setup

In this section, we describe the experimental setup that supports our investigation. In particular, we present the test collection and the retrieval baseline used to assess the effectiveness of UQEE. Additionally, we describe the experimental training and evaluation procedures.

Test Collections

To assess the effectiveness of our approach, we use the category B portion of ClueWeb09 [Clarke et al., 2009], a standard TREC web test collection with 50,220,423 documents, which has been extensively used in the information retrieval research field. We generate 50 queries using all the words from the title field of the corresponding TREC 2009 web track queries. The average query length is 2.1. Note that, 21 of the 50 queries mention at least one entity. As our basic retrieval system, we use Indri [Strohman et al., 2005], a language-model based search engine which provides

support for indexing and querying large corpora. The preprocessing of documents and queries included stemming with Porter's stemmer [Porter, 1980] and the removal of standard English stopwords.

As a knowledge base, we used the English Wikipedia. In particular, we built our entity repository W based upon a Wikipedia dump from June 1st, 2012. From this dump, we extracted a total of 2,069,704 unique entities, referred to by a total of 5,521,403 aliases. On average, this amounts to around 2.67 alternative aliases per entity.

Retrieval Baselines

UQEE can be implemented over any standard retrieval model. In our experiments we implemented it on top of the initial ranking produced by the BM25 retrieval model, reporting different results, one for each entity field described in Table 3.1.

Particularly, for each input query, we retrieve 1,000 documents using the BM25 retrieval model. This is our standard retrieval baseline, henceforth referred to as BM25. On top of the initial baseline ranking produced by BM25, we compare three instances of our unsupervised approach, one for each field, to the BM25 baseline, which performs no expansion. We refer to each instance of our approach as UQEE-IR, for the infobox relationship field, UQEE-IP, for the infobox property field, and UQEE-CO, for the content field. Additionally, in our experiments, we select the top k terms related to an entity recognized in the query to be used in the expansion, and we vary k from 10 to 10 up to 100, to investigate the retrieval performance when k increases.

Training and Evaluation Procedures

In order to ensure a fair assessment of UQEE and the BM25 baseline, we perform a 5-fold cross validation for the test collection described previously. In particular, for each cross-validation round, we train on four folds and test on the remaining fold. Accordingly, we report our results as an average across the test queries in each round, hence ensuring a complete separation between training and test queries at all times.

Regarding the evaluation of the investigated approaches, we report retrieval effectiveness in terms of two evaluation metrics: mean average precision (MAP), and precision at 10 (P@10). In particular, both MAP and P@10 are based on binary assessments of relevance. While MAP has been traditionally used for retrieval evaluation [Baeza-Yates and Ribeiro-Neto, 2011], P@10 is a typical target for web search evaluation, by focusing on the retrieval performance at early ranks [Jansen et al., 2000].

3.5.2 Results

In this section, we describe the experiments we have carried out to evaluate UQEE. In particular, we address the four research questions stated in Section 3.5, by contrasting the effectiveness of different instances of UQEE between them and to the BM25 baseline, described in Section 3.5.1. Significance is verified with a two-tailed paired t-test [Jain, 1991], with the symbol ▲ (▼) denoting a significant increase (decrease) at the $p < 0.05$ level, and the symbol • denoting no significant difference.

Infobox and Features Effectiveness

In this section, we address our first and second research questions, by assessing the effectiveness of the features, described in Section 3.4.1, used by UQEE to score candidate expansion terms extracted from the infobox property and content fields. To this end, Figure 3.4 shows the retrieval performance of each one of the considered features when deployed in isolation, in order to select the top k expansion terms for each query. In addition, we also report the retrieval performance of the two strategies to combine rankings, described in Section 3.4.2. As a baseline for this investigation, we include the performance of the BM25 retrieval model, which performs no expansion.

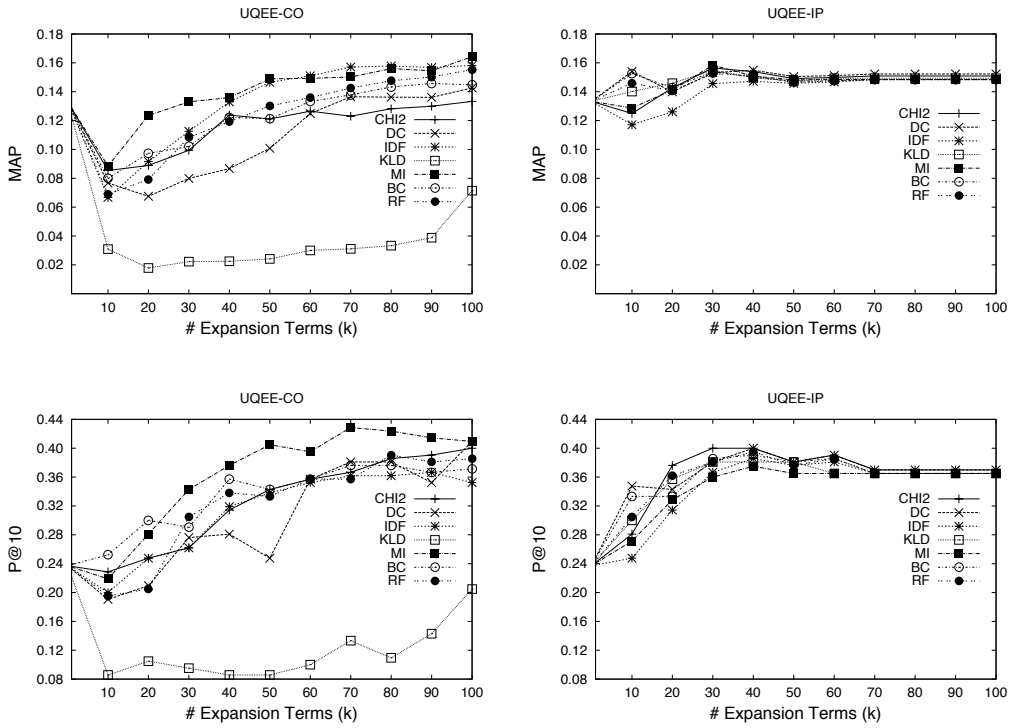

Figure 3.4. The effectiveness of features and ranking combination strategies.

Particularly, we measure the retrieval performance of the top-k candidate terms, selected using each feature and ranking combination strategy, with k varying from 10 to 100. This procedure was executed for the UQEE-CO and UQEE-IP instances, which consider the content and the infobox property fields, respectively. Note that, the retrieval performance when $k = 0$ corresponds to the BM25 baseline.

Figure 3.4 shows that UQEE-IP leads to better retrieval results than UQEE-CO for $k \leq 60$. Indeed, for $k \leq 40$, the difference between them is even greater. This is an important observation since, in a search system, shorter queries are preferable because they take less time to process, i.e., lower k leads to faster query processing. For $k > 60$, UQEE-CO outperforms UQEE-IP in some cases, because for only about 28% of the queries we have more than 60 candidate terms when using the infobox property field.

Additionally, the results for UQEE-IP reach their best values when $k = 30$, and remain stable from this point. Differently, the best results for UQEE-CO are achieved using $k = 100$ expansion terms. In particular, shorter queries generated by UQEE-IP and longer queries generated by UQEE-CO are roughly equivalent in terms of retrieval performance, but UQEE-IP provides lower query latency. Recalling our first research question, these observations attest the effectiveness of Wikipedia infoboxes for query expansion, showing that they effectively balance retrieval performance and query latency.

Figure 3.4 also shows that all features achieve positive results when used within UQEE-IP and UQEE-CO. Particularly, in UQEE-IP, all the features have a similar performance, with a slight advantage to DC and MI in all cases, while in UQEE-CO the MI feature significantly outperforms the others. The exception is the UQEE-CO with KLD feature, which presents poor performance, even when compared to the baseline. Particularly, the KLD feature overemphasizes the importance of the category for the term score, which contributes to the negative result. Recalling our second research question, these observations demonstrate the suitability of our features to select effective expansion terms.

Query Expansion Effectiveness

In this section, we address our third research question, by assessing the effectiveness of our unsupervised entity-oriented query expansion approach. To this end, Table 3.2 extends the results presented in Figure 3.4, which has already attested the effectiveness of our approach, and shows the overall retrieval performance of different instances of UQEE compared to the BM25 baseline, which performs no expansion.

In particular, the instances UQEE-CO30, UQEE-IP30, and UQEE-IR30 corre-

spond to the instances UQEE-CO, UQEE-IP, and UQEE-IR referred in Section 3.5.1, considering the use of the top $k = 30$ terms for query expansion, and the MI feature. Note that the UQEE-IR instance generates only a few expansion terms and there is no need to use features or ranking combination strategies to select some of them. Similarly, the instance UQEE-CO100 corresponds to the instance UQEE-CO, considering the use of the top $k = 100$ terms for query expansion, and the MI feature. These instances were chosen because they present the best retrieval performance as we can observe in Figure 3.4, except for the instance UQEE-CO30, which was chosen for comparative purposes.

Table 3.2. The overall retrieval performance of UQEE instances.

Approaches	MAP	P@10
BM-25	0.1333	0.2371
UQEE-CO30	0.1330 (-0.22%) $^{\bullet}$	0.3429 (+44.62%) $^{\blacktriangle}$
UQEE-IR30	0.1571 (+17.85%) $^{\blacktriangle\blacktriangle}$	0.3600 (+51.83%) $^{\blacktriangle\bullet}$
UQEE-IP30	0.1584 (+18.82%) $^{\blacktriangle\blacktriangle\bullet}$	0.4000 (+68.70%) $^{\blacktriangle\blacktriangle\blacktriangle}$
UQEE-CO100	**0.1642 (+23.18%)** $^{\blacktriangle\blacktriangle\bullet\bullet}$	**0.4095 (+72.71%)** $^{\blacktriangle\blacktriangle\blacktriangle\bullet}$

For all instances, the percentage improvement compared to the BM25 baseline is also shown. In addition, a first significance symbol denotes whether the improvements are statistically significant. For the UQEE-IR30, UQEE-IP30, and UQEE-CO100 instances, a second such symbol denotes significance with respect to UQEE-CO30. For the UQEE-IP30, and UQEE-CO100 instances, a third such symbol denotes significance with respect to UQEE-IR30. Finally, for UQEE-CO100, a forth symbol denotes significance compared to UQEE-IP30. The best value in each column is highlighted in bold.

From Table 3.2, we first observe that all instances of UQEE, except for UQEE-CO30, significantly improve upon the BM25 baseline. In particular, the gains are up to 23.18% in terms of MAP, and 72.71% in terms of P@10. Recalling our third research question, these observations attest the effectiveness of our unsupervised entity-oriented query expansion approach. Additionally, we also observe that UQEE-CO100 outperforms the other UQEE instances. However, the results show that the gains of UQEE-CO100 compared to UQEE instances that use information from infoboxes, specially UQEE-IP30, are negligible. Note that, UQEE-CO100 uses $k = 100$ expansion terms while UQEE-IP30 and UQEE-IR30 uses only $k = 30$ expansion terms. As mentioned before, in a search system, shorter queries are preferable because they take less time to process. These observations demonstrate that UQEE-IP and UQEE-IR instances successfully leverage the semantics of entities implicitly encoded in Wikipedia, once again attesting the effectiveness of Wikipedia infoboxes for query expansion.

Effectiveness of Automatically Generated Infoboxes

In this section, we address our fourth research question, by assessing the effectiveness of automatically generated infoboxes for query expansion. To this end, Table 3.3 shows the retrieval performance of the UQEE-IP30A, an instance of UQEE which corresponds to the instance UQEE-IP referred in Section 3.5.1, considering the use of the top $k = 30$ terms for query expansion using the MI feature, and the automatically generated infobox property field. Particularly, for all Wikipedia articles with infoboxes used by previous UQEE-IP instances, we discard the manually generated infoboxes, and automatically generate new ones using the WAVE approach proposed in Chapter 4. We then use this new infoboxes as a source of candidate expansion terms.

In addition, for an easy comparison, Table 3.3 also shows the retrieval performance of the UQEE-CO30, UQEE-IP30 and BM25 baselines, already presented in Table 3.2. For all instances, the percentage improvement compared to the BM25 baseline is also shown. In addition, a first significance symbol denotes whether the improvements are statistically significant. For the UQEE-IP30 and UQEE-IP30A instances, a second such symbol denotes significance with respect to UQEE-CO30. Lastly, for UQEE-IP30A, a third symbol denotes significance compared to UQEE-IP30. The best value in each column is highlighted in bold.

Table 3.3. The retrieval performance using infoboxes generated by WAVE.

Approach	MAP	P@10
BM-25	0.1333	0.2371
UQEE-CO30	0.1330 (-0.22%) $^{\bullet}$	0.3429 (+44.62%) $^{\blacktriangle}$
UQEE-IP30	**0.1584 (+18.82%)** $^{\blacktriangle\blacktriangle}$	**0.4000 (+68.70%)** $^{\blacktriangle\blacktriangle}$
UQEE-IP30A	0.1568 (+17.63%) $^{\blacktriangle\blacktriangle\bullet}$	0.3886 (+63.89%) $^{\blacktriangle\blacktriangle\blacktriangledown}$

From Table 3.3, we first observe that UQEE-IP30A significantly improves upon the BM25 and UQEE-CO30 baselines with gains of up 17.89% in terms of MAP, and 63.89% in terms of P@10. Additionally, the performance of UQEE-IP30A is roughly equivalent to the performance of UQEE-IP30, which considers manually generated infoboxes. Recalling our fourth research question, these observations attest that automatically generated infoboxes are suitable for query expansion. Moreover, they are as effective as manually generated infoboxes.

3.6 Summary

This chapter introduced UQEE, a novel approach for the query expansion problem described in Chapter 2. Our unsupervised entity-oriented query expansion approach

extensively uses information available in Wikipedia infoboxes, and a set of taxonomic features to select effective expansion terms.

In Section 3.1, we described the on-line and off-line stages of UQEE's retrieval process, and its main components. Additionally, we discussed about the UQEE computational overhead in query time compared to standard pseudo-relevance feedback approaches. In Section 3.2, we described how UQEE uses Wikipedia in order to derive an entity repository. We started discussing about the processing of Wikipedia articles, next presenting the Wikipedia fields actually used by our approach as a source of candidate expansion terms. We also provided an illustrative example of a Wikipedia article to discuss how UQEE addresses entity multi-label and multi-class issues found in Wikipedia. In Section 3.3, a complete example of the entity resolution procedure was provided in order to describe how UQEE recognizes named entities in queries, and maps an occurrence of a named entity in a query to the corresponding entity in the repository. Additionally, we presented a simple and effective mechanism based on popularity to address the entity disambiguation problem. Moreover, we described an automatic machine learning mechanism used by UQEE to determine the most representative entity in a query, an important issue to avoid topic drift in query expansion.

In Section 3.4, we described the ranking procedure adopted by UQEE to rank candidate expansion terms from the most representative entity in a query. We first defined five features adapted from the literature used to score candidate expansion terms, and then the two strategies adopted to combine different rankings produced using each one of the features. Lastly, in Section 3.5, we presented the experimental methodology and evaluation procedures which made it possible to validate UQEE in contrast to a standard retrieval baseline. Experimental results attested the effectiveness of UQEE, further showing the effectiveness of Wikipedia infoboxes and taxonomic features for query expansion. In addition, we show that automatically generated infoboxes are as effective as manually generated ones for query expansion.

Recalling our thesis statement from Section 1.1, in this chapter, we showed that the use of multiple sources of semantic evidence on entities, specially infoboxes, to devise discriminative term features are effective for query expansion. In the next chapter, we introduce WAVE, our self-supervised approach to autonomously extract attribute-value pairs from Wikipedia articles. WAVE is a web extractor, paramount to support our entity-oriented query expansion approaches based on Wikipedia.

Chapter 4

Automatic Infobox Generation

Over time, Wikipedia became a significant source of relevant information, having been used in different information retrieval tasks, such as question answering [Higashinaka et al., 2007; Kaisser, 2008], query expansion [Li et al., 2007; Milne et al., 2007], multilingual information retrieval [Potthast et al., 2008], and text categorization [Banerjee et al., 2007; Wang et al., 2007]. Recently, some approaches for automatic extraction of attribute-value pairs from Wikipedia have been proposed [Suchanek et al., 2007; Nguyen et al., 2007; Auer and Lehmann, 2007; Wu and Weld, 2007; Wu et al., 2008; Hahn et al., 2010]. In particular, they use heuristics to exploit lexical and syntactic patterns, and probabilistic models to segment and label sequence data from Wikipedia.

Differently from previous approaches in the literature, we propose a self-supervised approach that takes advantage of the Wikipedia article structure, represented in a novel enriched plain text format, and uses a window based segmentation model to learn how to extract an unlimited number of non-predefined attribute-value pairs from the article, in order to automatically generate infoboxes. Our approach is self-supervised in the sense that it uses *a priori* available information to learn a baseline extractor, and the training proceeds repeatedly by using the decisions of the extractor at step s to train the extractor at step $s + 1$.

The remainder of this chapter describes our approach to automatically generate infoboxes for Wikipedia articles. In particular, Section 4.1 presents its general operation, extraction flow, and main components. Section 4.2 describes the Wikipedia processor component, responsible for extracting from a Wikipedia corpus a set of training data used by the other components. In addition, it presents an example of the different textual formats for Wikipedia articles, including the novel enriched plain text format. Section 4.3 describes the article and sentence classifiers, which associate articles to in-

fobox templates, and sentences within articles to attributes, respectively. Section 4.4 describes the filter component, responsible for selecting, for each attribute, the most appropriate sentence from the set of associated sentences. Section 4.5 describes how our approach effectively extracts sequences of terms from sentences and assigns them to values of attributes. Lastly, Section 4.6 presents the experimental methodology and evaluation procedures used to validate our approach in contrast to a state-of-the-art baseline from the literature.

4.1 Overview

In the previous Chapter 3, we presented experimental results that attested the effectiveness of Wikipedia infoboxes for query expansion. As shown in Table 3.3, both manually and automatically generated infoboxes are suitable for query expansion. However, a significant obstacle to extensively use infoboxes is the fact that only about 20% of Wikipedia articles have an infobox. In this section, we introduce our approach to automatically generate infoboxes for Wikipedia articles. In particular, we propose a self-supervised approach to extract attribute-value pairs from the content of Wikipedia articles called WAVE, an acronym for "Wikipedia Attribute-Value Extractor". Algorithm 1 illustrates the general operation of our approach.

Algorithm 1 Automatic Generation of Wikipedia Infoboxes.

Input: Wikipedia article A_e, and Wikipedia corpus W
Output: infobox I_e for A_e composed by a set of attribute-value pairs $\{\langle a_1, V_1\rangle,\ldots,\langle a_m, V_m\rangle\}$

1: Associate A_e to an infobox template T from W, composed by a set of attributes $\{a_1, \ldots, a_m\}$
2: Create a set P of m attribute-value pairs, each one corresponding to an attribute in T
3: **for** $i = 1 \rightarrow m$ **do**
4: Assign a set S of sentences extracted from the textual content of A_e to P_i
5: Select the most representative sentence $S_k \in S$
6: Segment the sentence S_k into a sequence Q of n-words
7: Select the most representative subsequence B of Q
8: $V_i \leftarrow B$
9: $I_e \leftarrow P$

Given a Wikipedia article A_e and a Wikipedia corpus W, our approach first classifies A_e into a category represented by an infobox template T from W (line 1). Next, it creates a set P of attribute-value pairs, each one corresponding to an attribute specified in T by Wikipedia users (line 2). Then, our approach extracts sentences from

the textual content of A_e and, for each attribute specified in T, it: (i) assigns a set of sentences S (line 4); (ii) selects the most representative sentence in S, which is a sentence with concepts strongly related to the attribute (line 5); (iii) segments the sentence into a word sequence (line 6) and selects the most representative subsequence of words, which is the subsequence with words most strongly related to the attribute (line 7); (iv) associates the subsequence of words to the value of the attribute (line 8). Finally it produces the infobox for A_e by getting the set of generated attribute-value pairs (line 9).

Figure 4.1 shows the main components of WAVE that implement the approach described by Algorithm 1. In the following, we describe each one of the main components. Section 4.2 presents the Wikipedia processor component (step 1 in Figure 4.1) used to create training data for the classifier and extractor components. Section 4.3 introduces the article classifier (step 2 in Figure 4.1) and the sentence classifier (step 3 in Figure 4.1) components. Section 4.4 describes the filter component (step 4 in Figure 4.1) responsible for filtering sentences, and selecting the most appropriate one for each infobox attribute. Finally, Section 4.5 presents the extractor component (steps 5 and 6 in Figure 4.1) which learns models to extract information from word sequences to compose values of attribute-value pairs, ultimately creating the infobox.

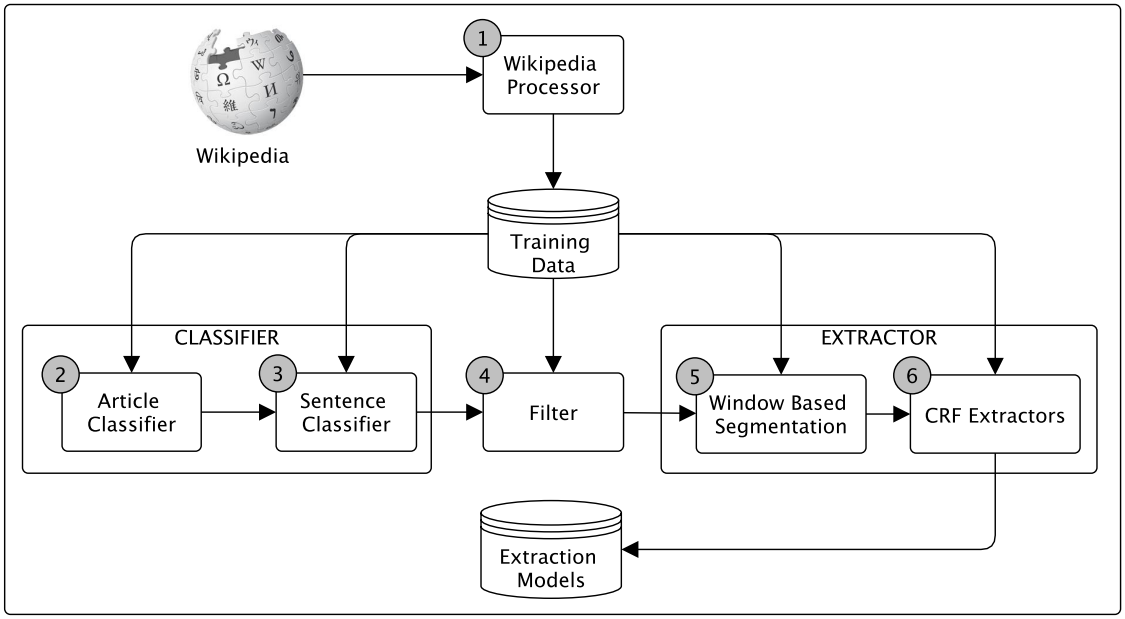

Figure 4.1. The main components of WAVE.

4.2 Processing Wikipedia Corpus

The Wikipedia processor component is responsible for extracting articles, infobox templates, and attribute-value pairs from a Wikipedia corpus. The extracted elements compose a set of training data used by the other components. The Wikipedia processor has five steps:

1. Scan the Wikipedia corpus and select articles associated with infobox templates.
2. Extract attribute-value pairs from the infoboxes within the selected articles and create infoboxes schemata. Note that, an infobox schema is a set of attributes for a specific infobox template.
3. Convert the content of each article from the Mediawiki text format, which follows the syntax used by Wikipedia users to edit articles, to HTML. For this, we use the Bliki engine, a Java API to parse Mediawiki content. Bliki engine is available at `http://code.google.com/p/gwtwiki/`.
4. Convert the content of each article from HTML to an *enriched plain text* format. The *enriched plain text* format considers only alpha-numeric, and punctuation characters, as well as HTML tags, but discarding from tags the attributes. Figure 4.2 shows an example of an article content in Mediawiki text format, HTML format, plain-text format and *enriched plain text* format.

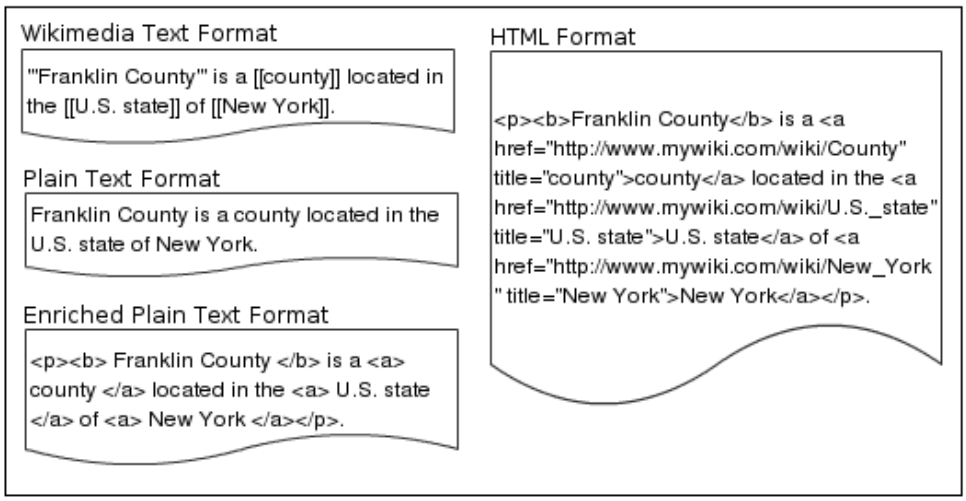

Figure 4.2. Example of a Wikipedia article in different text formats.

5. Segment the content of articles in *enriched plain text* format into sentences using OpenNLP, a Java API for natural language processing. OpenNLP is available at `http://opennlp.sourceforge.net/`.

Note that steps 3 and 4 are simply normalization procedures to enhance the syntactic regularity on the text.

4.3 Classifying Articles and Sentences

In the following, we describe the article classifier component, responsible to relate articles and infobox templates, and the sentence classifier component, responsible to relate article sentences and attributes.

4.3.1 Article Classifier

The Wikipedia processor described in Section 4.2 provides a set of categorized articles, i.e., articles related to infobox templates. This data is used by the article classifier to learn how to relate new articles and existing infobox templates. Each new article must be related to exactly one infobox template in order to determine which attributes should be extracted from the article content. We implement the article classifier using LIBSVM [Chang and Lin, 2011], a library for Support Vector Machines (SVM) [Boser et al., 1992; Cortes and Vapnik, 1995], which presents effective prediction performance for text categorization [Dumais et al., 1998]. In addition, we use infobox template names, article titles, the content of articles, and article categories as features for the article classifier.

4.3.2 Sentence Classifier

The processing of the sentence classifier can be divided into two phases:

1. Training phase: Data provided by the Wikipedia processor component described in Section 4.2 is used to build training data for the sentence classifier. For each article, sentences are related to attributes within the infobox schema of the article. The association is based on simple term matching. Terms within any value of an attribute which exactly match terms within any sentence will be related.

2. Learning phase: A maximum entropy classifier [Nigam et al., 1999] learns how to relate sentences and attributes based on training data generated in the previous phase. We use the OpenNLP Maxent Library, a framework for integrating information from many heterogeneous information sources for classification, to implement our classifier. It is known as a quite competitive alternative for multi-

class and multi-label classification. The OpenNLP Maxent Library is available at `http://maxent.sourceforge.net/`.

When a new article arrives, it is segmented into sentences. Then, the classification model learned by the sentence classifier is applied and article sentences are related to article attributes.

4.4 Filtering Sentences

The sentence classifier described in Section 4.3.2 is a multi-class classifier and can relate a set of sentences to the same attribute. The filter component is responsible for choosing the most appropriate sentence in the set.

Considering that we have the same attribute in several infobox schema in training data, we take all the sentences related to an attribute and group them into clusters using an efficient implementation of the k-means clustering algorithm [Kanungo et al., 2002]. To compute the distance between sentences, we represent them in a vector space and use the similarity measure, as defined by the vector space model [Baeza-Yates and Ribeiro-Neto, 2011]. From the clusters, we select the one with the largest number of sentences which come from different infobox schema. This heuristic selection is based on the intuition that the most popular cluster tends to be the one which contains the best value (the best sentence) to be related to the processed attribute. Then, we use the proximity to the cluster centroid to choose one sentence for each infobox schema. The sentence closest to the centroid is considered as the best option to fill the value of the attribute.

4.5 Extracting Values for Attributes

The extractor is responsible for extracting term sequences from text segments and to relate these sequences to values of attributes. Next, we present the window-based segmentation, and the CRF extractor components.

4.5.1 Window-Based Segmentation

There is a preprocessing procedure that must be done in an attempt to maximize conditional random fields extractors performance. Each sentence related to each attribute must be segmented into terms and a term sequence must be selected to compose the text segment to be processed by a conditional random fields extractor.

Particularly, for each sentence filtered by the filter component described in Section 4.4, it is possible to determine, with the same term matching procedure used by the sentence classifier described in Section 4.3.2, the terms of the sentence which correspond to the value of the attribute. We call these terms *attribute-terms*. Using a window size of x, we can extract, from each sentence, a term sequence composed by x terms before the attribute-terms (*pre-terms*), the attribute-terms, and x terms after the attribute-terms (*post-terms*). The extracted term sequences compose a training data for segmentation. Note that, the value of x must be empirically obtained.

When a new sentence arrives, it is segmented into terms. Then, we use the similarity between the *pre-terms* and *post-terms*, extracted from the sentence, and the *pre-terms* and *post-terms* in the training data, to select which terms will be used by the CRF Extractor.

4.5.2 CRF Extractor

Extracting values of attributes from a text can be viewed as a sequential data-labeling problem. Therefore, the choice of conditional random fields to address the problem is feasible, since conditional random fields is the state-of-the-art for this task. Our approach trains a different CRF extractor for each attribute, using a well-known implementation of conditional random fields available at `http://crf.sourceforge.net/`. For each attribute, the term sequences related to it by the window based segmentation component described in Section 4.5.1 are used to train the extractor. We label the *pre-terms* with the *pre* label, the *post-terms* with the *post* label, and each one of the *attribute-terms* with three different types of labels:

1. If the *attribute-terms* are composed by only one term, this term is labeled with the *init* label.

2. If the *attribute-terms* are composed by only two terms, the first term of the sequence is labeled with the *init* label and the last one is labeled with the *end* label.

3. If the *attribute-terms* are composed by more than two terms, the first term of the sequence is labeled with the *init* label, the last one is labeled with the *end* label, and each one of the other terms is labeled with the *middle* label.

Each one of the CRF extractors learns a different extraction model and uses it to extract values from term sequences. The extracted value is assigned to the attribute generating an attribute-value pair.

4.6 Experiments

In order to validate WAVE, we contrast it to KYLIN [Wu and Weld, 2007], the state-of-the-art baseline, across representative datasets. In particular, we aim to answer the following research question: How effective is our approach to automatically generate infoboxes?

4.6.1 Setup

In this section, we describe the experimental setup that supports our investigation. In particular, we present the datasets used to assess the effectiveness of our approach, and we describe the training and evaluation procedures used in our experiments.

Datasets

To assess the effectiveness of our approach, we created four datasets based upon a Wikipedia dump from December 3rd, 2010, one for each of the following popular Wikipedia infobox templates: *U.S. County*, *Airline*, *Actor*, and *University*. Table 4.1 shows the distribution of the 3,610 Wikipedia articles in the four datasets. The size of the datasets is the number of Wikipedia articles within them.

Table 4.1. The distribution of Wikipedia articles in the datasets.

	U.S. County	Airline	Actor	University
Dataset size	1,697	456	312	1,145

For each dataset, we consider a set of attributes extracted directly from infoboxes within its articles. However, some attributes do not occur frequently in Wikipedia articles, therefore being discarded. We discarded all attributes not present in at least 15% of the articles in each dataset. Furthermore, the sentence classifier component also discards attributes from the Wikipedia articles, when the attribute values do not match any word sequence within the article sentences. Table 4.2 shows the number of attributes extracted from the Wikipedia articles, the number of attributes discarded due to low frequency and the matching procedure, and the total number of attributes actually used in each dataset.

There are three different types of attribute in each dataset: date, number, and string. Table 4.2 shows that we used 54 different attributes in the experiments. The most common type of attribute is string (55.55%), followed by number (27.78%), and date attributes (16.67%). Figure 4.3 shows the distribution of the attribute types in the datasets, with string in black, date in gray and number in white. From it, we

Table 4.2. Extracted, discarded, and used attributes for datasets.

Dataset	Number of Attributes			
	Extracted	Discarded		Used
		Low frequency	No matching	
U.S. County	105	75 (71.42%)	11 (10.48%)	19 (18.10%)
Airline	60	40 (66.67%)	9 (15.00%)	11 (18.33%)
Actor	45	34 (75.55%)	6 (13.34%)	5 (11.11%)
University	283	251 (88.70%)	13 (4.59%)	19 (6.71%)

observe that only in the *U.S. County* dataset the string type is not majority. In this case, the number of numerical attributes is greater than the number of attributes of the string type.

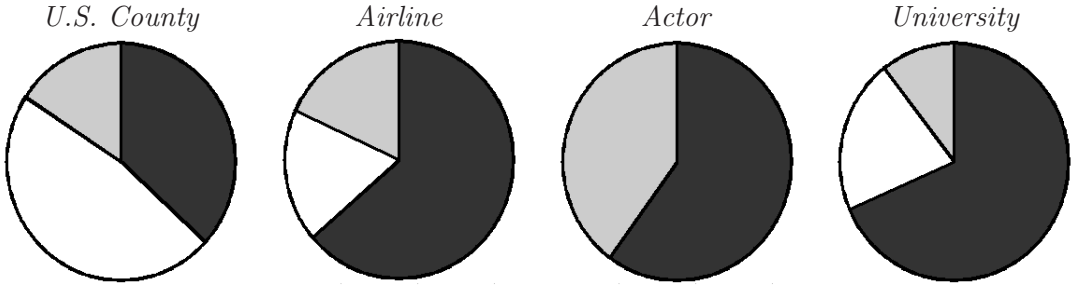

Figure 4.3. The distribution of attribute types per dataset.

Training and Evaluation Procedures

In order to ensure a fair assessment of WAVE and KYLIN, we perform a 5-fold cross validation (Stone [1974]) for the datasets described in previous section. Particularly, for each cross-validation round, we train on four folds and test on the remaining fold. Accordingly, we report our results as an average across each round, hence ensuring a complete separation between training and test at all times.

Regarding the evaluation of the investigated approaches, we report effectiveness in terms of three evaluation metrics: precision (P), recall (R), and $F1$. In particular, P is defined as the proportion of correctly extracted attribute-value pairs in the set of all extracted attribute-value pairs, R is defined as the proportion of correctly extracted attribute-value pairs in all of the correctly attribute-value pairs in the examples, and $F1$ is a combination of P and R, defined as $2PR/(P+R)$.

Additionally, we performed preliminary experiments in order to empirically obtain the value of the window size x used by the window based segmentation procedure described in Section 4.5.1. We use $x = 3$ in all experiments. Finally, as mentioned before, WAVE trains a different CRF extractor for each attribute. Thus, in order to evalu-

ate the effectiveness of the extraction process, we perform a 10-fold cross validation, computing the average precision, average recall, and average F1, for each attribute.

4.6.2 Results

In this section, we describe the experiments we have carried out to evaluate WAVE. In particular, we address the research question stated in the beginning of Section 4.6, by contrasting the effectiveness of WAVE to the KYLIN baseline. Significance is verified with a two-tailed paired t-test [Jain, 1991] at the $p < 0.05$ level.

Overall Effectiveness

In this section, we address our research question by assessing the effectiveness of our approach. To this end, Table 4.3 shows the overall extraction performance for WAVE and the baseline for the datasets described in Section 4.6.1. The values of P, R, and $F1$ correspond to the average precision, average recall and average F1 for the group of attributes within each dataset. The percentage improvement compared to the KYLIN baseline are also shown. In addition, the best value for each evaluation metric is highlighted in bold.

Table 4.3. Extraction performance (per dataset).

U.S. County			
	P	R	F1
KYLIN	0.8773	0.8640	0.8701
WAVE	0.9385 (+6.97%)	0.9392 (+8.71%)	0.9387 (+7.88%)
Airline			
	P	R	F1
KYLIN	0.5573	0.4975	0.5240
WAVE	**0.6795 (+21.92%)**	**0.6311 (+26.86%)**	**0.6513 (+24.29%)**
Actor			
	P	R	F1
KYLIN	0.6582	0.6057	0.6309
WAVE	0.7531 (+14.42%)	0.6873 (+13.48%)	0.7160 (+13.53%)
University			
	P	R	F1
KYLIN	0.6387	0.5612	0.5927
WAVE	0.7159 (+12.08%)	0.6333 (+12.83%)	0.6659 (+12.34%)

From Table 4.3, we observe that WAVE significantly outperforms the baseline. In particular, the gains are up to 21.92% in terms of precision, 26.86% in terms of recall, and 24.29% in terms of F1. Note that the gains are greater in datasets with more string and date attributes, an expected result, since the baseline already presents high quality results for numerical attributes, as we can observe in Table 4.4. Furthermore, the type

string takes more advantage of the textual content enrichment made by WAVE. Remember that WAVE enriches the textual content of the articles with HTML structural information, making word patterns more regular, which increases the performance of the CRF extractors. Recalling our research question, these observations attest the effectiveness of our approach for autonomously extracting attribute-value pairs from Wikipedia articles.

4.6.2.1 Effectiveness for Attribute Types

In this section, we extend the analysis described in the previous section, by assessing the effectiveness of WAVE, considering different types of attributes. To this end, Table 4.4 shows the performance of WAVE and KYLIN for the types of attributes described in Section 4.6.1. The values of P, R, and $F1$ correspond to the average precision, average recall and average F1, for the group of attributes of each type throughout the datasets. The percentage improvements compared to the KYLIN baseline are also shown.

Table 4.4. Extraction performance (per attribute type).

Approach	Date			Number			String		
	P	R	F1	P	R	F1	P	R	F1
KYLIN	0.5746	0.5234	0.5467	0.8919	0.8816	0.8858	0.6258	0.5530	0.5840
WAVE	0.6333 10.20%	0.5734 9.56%	0.6004 9.82%	0.9234 3.53%	0.9304 5.54%	0.9264 4.58%	0.7437 18.85%	0.6705 21.25%	0.7002 19.91%

From Table 4.4, we observe that WAVE improvements are more pronounced for string attributes. In particular, the gains are 18.85% in terms of precision, 21.25% in terms of recall, and 19.91% in terms of F1, considering this attribute type. As mentioned before, string attributes take more advantage of the textual content enrichment provided by WAVE. This occurs because string attributes present more regular word patterns, and the greater the regularity in the word patterns of the HTML tags around the attribute value to be extracted within sentences, the better the performance of the CRF extractors. Moreover, the gains for numerical attributes are smaller, but still significant, since the results of the baseline in this case are already high, with almost no room for improvement.

4.7 Summary

This chapter introduced WAVE, a novel approach to address the automatic infobox generation problem. Our self-supervised approach autonomously extracts attribute-value pairs from a Wikipedia article by first representing it in a novel enriched plain

text format, next using a window based segmentation model to learn how to extract the attribute-value pairs from this article representation.

Throughout Sections 4.1 to 4.5, we described the general operation of WAVE. Particularly, in Section 4.1, we presented the algorithm that describes the extraction flow of WAVE. Additionally, we described its main components. In Section 4.2, we described the first WAVE component, which process a Wikipedia corpus to extract training data used by the other components. We also introduced our proposed enriched plain text format to represent Wikipedia articles. In Section 4.3, we described the WAVE classifiers. In particular, we described a classifier used to relate articles and infobox templates, and another one used to relate article sentences and attributes. In Section 4.4, we described the filter component, used to select, for each attribute, a single article sentence related to it. From this article sentence, our extractors take values for the attribute. In Section 4.5, we described how the sentence segmentation is done in order to enable our CRF extractors to extract sequences of terms, ultimately assigning them to values of attributes.

Finally, in Section 4.6, we presented the experimental methodology and evaluation procedures, which made it possible to validate WAVE in contrast to a state-of-the-art baseline. Experimental results attested the effectiveness of our self-supervised approach for automatic infobox generation, with gains of up to 21.92% in terms of precision, 26.86% in terms of recall, and 24.29% in terms of F1 over the baseline. Moreover, we observed that WAVE's performance is sensitive to the word pattern regularity of the tags around the values to be extracted from article sentences. Most important, the unsupervised entity-oriented query expansion approach presented in Chapter 3 used WAVE to automatically generate infoboxes, which were shown to be as effective as manually generated infoboxes.

In the next chapter, we markedly extend our unsupervised approach for query expansion described in Chapter 3 by introducing a novel learning to rank approach, which combines multiple features to rank candidate expansion terms. Particularly, this approach exploits the strengths and weaknesses of past research in order to deliver an effective solution for the query expansion problem.

Chapter 5

Supervised Entity-Oriented Query Expansion

Extending our unsupervised entity-oriented query expansion approach described in Chapter 3, we propose a supervised learning approach, which not only considers semantic evidence encoded in the content of Wikipedia article fields, but also exploits novel discriminative features, such as each term's distribution across multiple article fields, and its proximity to the original query terms.

In contrast to existing supervised approaches in the literature [Xu et al., 2009; Lin et al., 2011], our approach does not rely on a binary classification of candidate expansion terms as either useful or non-useful. Instead, it tackles query expansion as a learning to rank problem, in order to directly learn an effective ranking of the candidate terms related to an entity in the query. As a result, not only does it choose effective terms for expansion, but it also learns how to weigh their relative importance in the expanded query.

The remainder of this chapter describes our supervised entity-oriented query expansion approach. In particular, Section 5.1 introduces our approach, describing its retrieval process, and main components. Section 5.2 succinctly discusses the entity representation and resolution procedures, since they are similar to the corresponding procedures of our unsupervised approach described in Sections 3.2 and 3.3. We focus on the differences between them, specially in the article fields considered as a source of candidate expansion terms. Section 5.3 describes our learning to rank approach, and defines the features used to instantiate it within our supervised approach. Lastly, Section 5.4 presents the experimental methodology and evaluation procedures used to validate our approach in contrast to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches.

5.1 Overview

As mentioned in Section 3.1, the presence of a named entity in a query provides an opportunity for web search engines to improve their understanding of the user's information need. In this section, we introduce our supervised entity-oriented query expansion approach called L2EE, an acronym for "Learning To Expand using Entities". The retrieval flow of L2EE is illustrated in Figure 5.1.

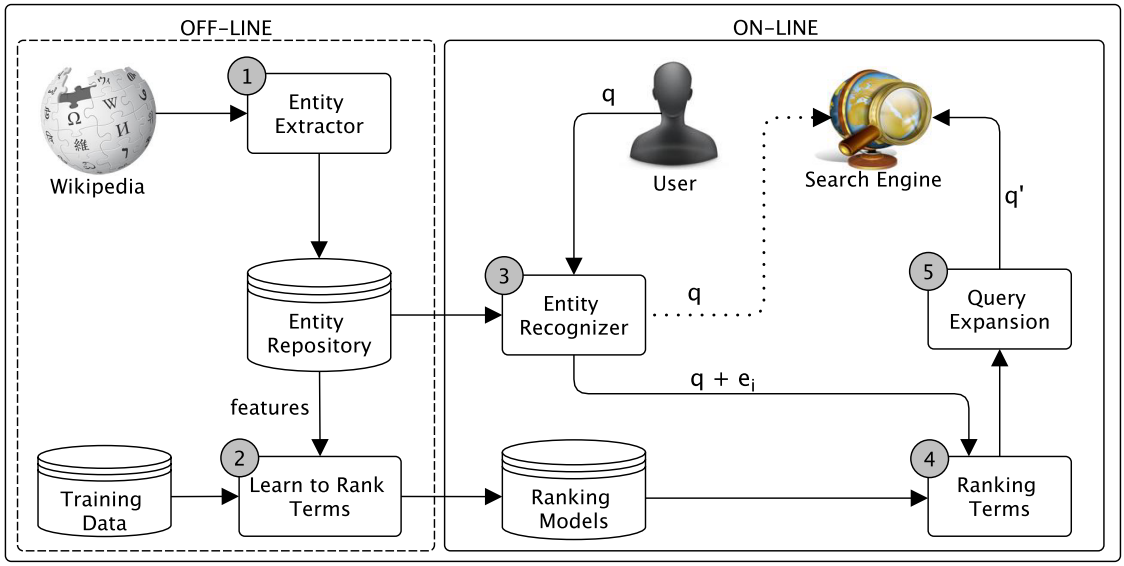

Figure 5.1. The off-line and on-line query processing with L2EE.

Both off-line and on-line stages of L2EE are very similar to UQEE described in Section 3.1. In the off-line stage, L2EE also assembles an entity repository W by processing a knowledge base (step 1 in Figure 5.1). In addition, given suitable training data, this stage is also responsible for learning the ranking function that will be used for identifying effective expansion terms related to entities in the repository (step 2). In the on-line stage, L2EE also generates a new expanded query q' based on the given user's query q by recognizing in q a named entity e_i from the repository W (step 3 in Figure 5.1). Candidate terms related to e_i, as recorded in the repository W, are ranked with respect to their predicted effectiveness given the query q (step 4), using the ranking function learned off-line in step 2. Lastly, the top k ranked terms according to the learned function are appended to q in order to produce the expanded query q' (step 5), which will be then used by the search engine to retrieve the final ranking of results to be presented to the user.

As with UQEE, most of the work of L2EE is done at the off-line stage, and the computational cost of the on-line stage is negligible. The computational overhead of

L2EE at query time is also lower than standard pseudo-relevance feedback approaches because, as well as in UQEE, only the modified query is processed.

5.2 Entity Representation and Resolution

Similarly to UQEE, our supervised entity-oriented query expansion approach builds an entity repository W using Wikipedia. Particularly, each field in F_i also comprises textual content from a specific region in the article that describes the entity e_i, but L2EE considers more fields than UQEE. Table 5.1 presents the article fields considered by our supervised entity-oriented query expansion approach.

Table 5.1. The article fields considered by L2EE.

Field	Description
title	the title of the article (unique identifier)
summary	the article's main concepts
infobox	special tabular structure that presents a set of attribute-value pairs describing different aspects of the article
category	categories used by Wikipedia users to classify the article
link	anchor-text from other articles in Wikipedia with a hyperlink to the article
appendix	external sources of information about the article, such as references and further reading
content	textual content of the remaining fields

In order to improve the recognition of named entities in web search queries, L2EE also uses multiple names as the set of aliases A_i of the entity e_i, and also exploits infobox templates as a means to identify the single most representative class of an article. Lastly, the entity resolution step of L2EE is identical to the entity resolution step of UQEE, described in Section 3.3.

5.3 Ranking Entity Terms

In order to rank effective expansion terms related to the most representative entity identified in the user's query, we introduce a learning to rank approach. In the remainder of this section, we formalize this approach and describe the features that are used to instantiate it in our experiments.

5.3.1 Learning a Ranking Model

In order to tackle query expansion as a ranking problem, we follow the general framework of discriminative learning [Liu, 2009]. In particular, our goal is to learn an optimal hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$, mapping the input space $\mathcal{X}$ to the output space $\mathcal{Y}$. To this

end, a plethora of machine learning algorithms could be deployed. In this work, we adopt a pairwise learning to rank formulation, which reduces the ranking problem to a binary classification problem [Liu, 2009], namely, that of choosing the most effective expansion term from a pair of candidate terms. We used the linear RankSVM algorithm [Joachims, 2006] to implement this pairwise learning to rank formulation. As a result, our input space $\mathcal{X}$ comprises pairs of learning instances of the form (x_u, x_v), where each instance x conveys a vector representation $\Phi(q, t)$ of a candidate expansion term t for a given query q, according to the feature extractor Φ. The various features considered in this work are described in Section 5.3.2. In order to guide the learning process towards identifying *effective* expansion terms, we consider an output space $\mathcal{Y}$ comprising binary performance-oriented labels y_{uv}, defined as:

$$y_{uv} = \begin{cases} -1 & \text{if } \delta(t_u) < \delta(t_v), \\ +1 & \text{otherwise,} \end{cases} \tag{5.1}$$

where $\delta(t)$ measures the gain attained by appending the candidate expansion term t to the query q, according to:

$$\delta(t) = \frac{\epsilon(q \cup \{t\}) - \epsilon(q)}{\epsilon(q)}, \tag{5.2}$$

where ϵ can be any standard information retrieval evaluation metric, such as mean average precision (MAP) [Baeza-Yates and Ribeiro-Neto, 2011].

Given m training queries $\{q_i\}_{i=1}^{m}$, their associated pairs of candidate expansion terms $(x_u^{(i)}, x_v^{(i)})$, and the label $y_{uv}^{(i)}$ associated with each pair, our goal is to learn a hypothesis h that minimizes the empirical risk $R(h)$, according to:

$$R(h) = \frac{1}{m} \sum_{i=1}^{m} \Delta(y_{uv}^{(i)}, h(x_u^{(i)}, x_v^{(i)})), \tag{5.3}$$

where the loss function Δ quantifies the penalty incurred by predicting an output $h(x_u^{(i)}, x_v^{(i)})$ when the correct output is $y_{uv}^{(i)}$. In our experiments, the loss function Δ is defined in terms of the total number of swapped pairs $(x_u^{(i)}, x_v^{(i)})$.

Lastly, as candidate hypotheses, we consider linear functions $h(x_u^{(i)}, x_v^{(i)}) = w^T(x_u^{(i)} - x_v^{(i)})$, parametrized by a weight vector w. In particular, our goal is to find a vector w that minimizes the empirical risk in Equation (5.3). Given a learned weight vector w, we can predict the effectiveness of all candidate expansion terms associated with the most representative entity in an unseen query q. The top k among these

terms are then appended to q, with their predicted scores serving as their weight in the expanded query q'.

5.3.2 Ranking Features

To represent candidate terms in a suitable form for our learning to rank approach, we employ a total of five statistical descriptors as term features: Dice's coefficient (DC), mutual information (MI), term frequency (TF), term spread (TS), and term proximity (TP). Our first two features, DC and MI, are taxonomic features that take into account not only the relevance of a term to an entity, but also to the class to which the entity belongs. These features have been shown to be effective descriptors of terms in Wikipedia articles in our previous analytical study [Brandão et al., 2011]. The DC and MI features were already defined previously in Equation 3.1 and Equation 3.2, respectively.

Our next two features, TF and TS, are statistical measures that take into account the distribution of terms in different fields. In particular, TS was adapted from previous work in the literature [Fernandes et al., 2007], having been used for query expansion [Oliveira et al., 2012]. In order to formalize these features, let $freq(t, f_j)$ be the frequency of the term t in the field $f_j \in F_i$ of the entity e_i. The term frequency (TF) of term t can be defined as:

$$\mathrm{TF}(t) = \sum_{j=1}^{|F_i|} freq(t, f_j), \tag{5.4}$$

where $|F_i|$ denotes the total number of available fields for the entity e_i, as described in Table 5.1. Different from TF, the term spread (TS) feature measures the spread of a term across multiple fields, i.e., the number of different fields in which a term occurs, according to:

$$\mathrm{TS}(t) = \sum_{j=1}^{|F_i|} \mathbf{1}_{f_j}(t), \tag{5.5}$$

where $\mathbf{1}_{f_j}(t)$ is the indicator function, equal to one if $t \in f_j$, or zero otherwise. Intuitively, the higher the values of TF and TS, the more t is related to e_i.

Lastly, we devise a feature to measure the proximity between a candidate term t and the original query terms. Particularly, we define the term proximity (TP) feature as:

$$\mathrm{TP}(t) = \sum_{j=1}^{l} \sum_{w=1}^{m} \log \frac{\mathit{freq}(\langle t, t_j \rangle, w)}{2^{w-1}}, \tag{5.6}$$

where t_j is the j-th term of the query q, l is the total length of q, and $\mathit{freq}(\langle t, t_j \rangle, w)$ is the total number of occurrences of the (unordered) pair $\langle t, t_j \rangle$ within windows of size w sentences across the concatenation of all fields of the entity e_i. Note that $w = 1$ denotes an occurrence of t and t_j within the same sentence. We consider $m = 5$, since preliminary experiments show that for $m > 5$ the value of the feature does not change significantly.

5.4 Experiments

In order to validate our supervised entity-oriented query expansion approach, we contrast it to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches across multiple web test collections. In particular, we aim to answer the following research questions:

1. How effective is our supervised learning approach for query expansion?
2. How does our approach perform for entity queries?
3. How does our approach perform for difficult queries?
4. How effective are our approach for non-Wikipedia pages[1]?
5. Which features are effective for query expansion?

5.4.1 Setup

In this section, we describe the experimental setup that supports our investigation. In particular, we present the test collections and retrieval baselines used to assess the effectiveness of L2EE, we introduce an oracle expansion mechanism used to analyze the performance of our approach for difficult queries, as well as to assess the room for improvement, and finally we describe the training and evaluation procedures.

[1]We consider non-Wikipedia pages an instance of standard TREC web test collections without Wikipedia pages.

Test Collections

To assess the effectiveness of L2EE, we use three standard TREC web test collections: WT10g [Hawking and Craswell, 2001], GOV2 [Büttcher et al., 2006], and the category B portion of ClueWeb09 [Clarke et al., 2009], or simply CW09B. For each test collection, we generate queries using all the words from the title field of the corresponding search tracks at TREC. Table 5.2 summarizes salient statistics of these test collections. Besides the TREC tracks used in each collection, we describe the number of documents and queries, as well as the average query length, the number of entity queries (i.e., queries with at least one identified entity), and the number of difficult queries (see Section 5.4.2 for a precise definition).

Table 5.2. Overview of the considered standard TREC web test collections.

	WT10g	GOV2	CW09B
TREC track	Web 00/01	Terabyte 04/05/06	Web 09/10
# documents	1,692,096	25,205,179	50,220,423
# queries	100	149	98
avg. query length	4.18	3.10	2.06
# entity queries	98	148	94
# difficult queries	7	29	11

As well as in UQEE, Indri [Strohman et al., 2005] was used as the basic retrieval system for L2EE. Similarly, the preprocessing of documents and queries included stemming with Porter's stemmer [Porter, 1980] and the removal of standard English stopwords. Additionally, we also used the English Wikipedia as a knowledge base, building our entity repository W based upon a Wikipedia dump from June 1st, 2012.

Retrieval Baselines

Our supervised entity-oriented query expansion approach can be implemented over any standard retrieval model, such as BM25 and language models. In our experiments we implemented L2EE, as well as two state-of-the-art query expansion baselines, on top of the initial ranking produced by the Kullback-Leibler (KL) retrieval model.

In particular, for each input query, we retrieve 1,000 documents using the KL retrieval model with Dirichlet smoothing [Zhai and Lafferty, 2001a]. This formulation has been shown to be effective across many retrieval scenarios, and represents the current state-of-the-art in language modeling [Zhai, 2008]. In our experiments, the smoothing parameter μ of the Dirichlet prior was empirically set to 2,500, following the training procedure described in next sections.

On top of the initial baseline ranking produced by the KL retrieval model, we compare our learning to rank approach to two state-of-the-art query expansion baselines. Our first query expansion baseline is an implementation of Lavrenko's relevance models (RM1) [Lavrenko and Croft, 2001] provided by Indri, which instantiates the classical pseudo-relevance feedback approach to query expansion. In addition to RM1, we compare L2EE to the entity-oriented pseudo-relevance feedback approach of [Xu et al., 2009], henceforth referred to as QD. As discussed in Section 2.2.2, this approach represents the current state-of-the-art in entity-oriented query expansion, and is hence our strongest baseline. In our experiments, RM1 is used to select the top $k = 50$ terms from the top $n = 10$ documents retrieved by the KL retrieval model. As for the QD and L2EE query expansion approaches, both are deployed to select the top $k = 50$ terms related to an entity identified in the query.

In preliminary experiments, we varied k from 10 to 100 and, as expected, we observed that retrieval performance increases when k increases, i.e., greater k leads to better search results. However, greater k values imply worse time performance. As mentioned before, in search systems, shorter queries are preferable because they take less time to process, i.e., lower k leads to faster query processing. In our experiments, we set $k = 50$ to balance time and retrieval performance. This setting provided the best overall performance during training for the QD and L2EE query expansion approaches, since for $k > 50$ the retrieval effectiveness gains are negligible while time performance worsens.

Oracle

Query expansion approaches usually lead to global improvements compared to a baseline ranking that does not perform any expansion. Nevertheless, query expansion may also be harmful to some queries. This is particularly the case for difficult queries, i.e., queries with a poor first-pass retrieval performance, which end up returning irrelevant documents to be used as feedback [Amati et al., 2004]. In order to analyze the performance of our supervised entity-oriented query expansion approach for difficult queries, as well as to assess the room for improvement, we introduce an oracle expansion mechanism, which knows exactly whether or not to expand each individual query.

Given a query q with corresponding relevance assessments, our oracle mechanism begins by selecting $n = 10$ documents that are relevant to this query.[2] Each unique term t_i extracted from the feedback documents is then assessed as to the extent to

[2]For queries with more than $n = 10$ relevant documents, we break ties randomly.

which it improves the retrieval performance (in terms of MAP) of the query q, when appended to this query. After discarding non-improving terms, the remaining terms are appended to the query q. The improvement in MAP observed for each term t_i in the previous step is used as the weight w_i of the term in the new expanded query.

Our oracle expansion mechanism does not determine the best possible combination of terms and weights to expand queries. It approximates an optimal selection of terms greedily, by selecting one term at a time. While this simplified approach is indeed suboptimal, it provides a reasonably strong lower-bound of the optimal performance. More importantly for the feasibility of our investigations, it avoids the combinatorial selection of the single best set of terms, which may become prohibitive even with a few candidate terms.

Training and Evaluation Procedures

Most retrieval approaches investigated in this chapter require some form of supervised training. In order to ensure a fair assessment of these approaches, we perform a 5-fold cross validation for each of the test collections described previously. In particular, for each cross-validation round, we train on four folds and test on the remaining fold. Accordingly, we report our results as an average across the test queries in each round, hence ensuring a complete separation between training and test queries at all times.

Regarding the evaluation of the investigated approaches, we report retrieval effectiveness in terms of three evaluation metrics: mean average precision (MAP), normalized discounted cumulative gain (nDCG), and precision at 10 (P@10). Particularly, both MAP and P@10 are based on binary assessments of relevance, whereas nDCG can leverage graded relevance assessments. As mentioned in Section 3.5.1, while MAP has been traditionally used for retrieval evaluation [Baeza-Yates and Ribeiro-Neto, 2011], both nDCG and P@10 are typical targets for web search evaluation, by focusing on the retrieval performance at early ranks [Jansen et al., 2000].

5.4.2 Results

In this section, we describe the experiments we have carried out to evaluate our entity-oriented query expansion approach. In particular, we address the five research questions stated in Section 5.4.1, by contrasting the effectiveness of L2EE to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback baselines, as well as to an oracle expansion mechanism. Significance is verified with a two-tailed paired t-test [Jain, 1991], with the symbol ▲ (▼) denoting a significant increase (decrease) at the $p < 0.05$ level, and the symbol • denoting no significant difference.

Query Expansion Effectiveness

In this section, we address our first research question, by assessing the effectiveness of our supervised entity-oriented query expansion approach. To this end, Table 5.3 shows the retrieval performance of L2EE compared to KL, which performs no expansion, RM1, and QD. In order to provide a fair comparison to entity-oriented pseudo-relevance feedback approaches, both QD and L2EE fall back to a standard pseudo-relevance feedback approach for queries without named entities.[3] For all query expansion approaches (i.e., RM1, QD, and L2EE), percentage improvement figures compared to the KL baseline are also shown. In addition, a first instance of the aforementioned significance symbols denotes whether these improvements are statistically significant. For the entity-oriented pseudo-relevance feedback approaches (i.e., QD and L2EE), a second such symbol denotes significance with respect to RM1. Finally, for L2EE, a third symbol denotes significance compared to QD. The best value in each row is highlighted in bold.

Table 5.3. Retrieval performance (all queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.1953	0.2030 (+3.94%) $^{\bullet}$	0.2131 (+9.11%) $^{\blacktriangle\bullet}$	**0.2628 (+34.56%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2730	0.2840 (+4.03%) $^{\bullet}$	0.3130 (+14.65%) $^{\blacktriangle\bullet}$	**0.3740 (+37.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.4686	0.4693 (+0.14%) $^{\bullet}$	0.4924 (+5.07%) $^{\bullet\bullet}$	**0.5311 (+13.33%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.2947	0.3185 (+8.07%) $^{\bullet}$	0.3240 (+9.94%) $^{\blacktriangle\bullet}$	**0.3661 (+24.22%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.5416	0.5624 (+3.84%) $^{\bullet}$	0.6047 (+11.65%) $^{\blacktriangle\bullet}$	**0.6866 (+26.77%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.5860	0.6014 (+2.62%) $^{\bullet}$	0.6132 (+4.64%) $^{\blacktriangle\bullet}$	**0.6418 (+9.52%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.1416	0.1429 (+0.91%) $^{\bullet}$	0.1648 (+16.38%) $^{\blacktriangle\bullet}$	**0.1820 (+28.53%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2408	0.2500 (+3.82%) $^{\bullet}$	0.3082 (+28.00%) $^{\blacktriangle\blacktriangle}$	**0.3663 (+52.12%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3720	0.3605 (-3.09%) $^{\bullet}$	0.3951 (+6.20%) $^{\bullet\blacktriangle}$	**0.4132 (+11.07%)** $^{\blacktriangle\blacktriangle\bullet}$

From Table 5.3, we first observe that L2EE significantly improves upon all baselines, and across all three test collections. In particular, compared to KL, the gains are up to 34.56% in terms of MAP, 52.12% in terms of P@10, and 13.33% in terms of nDCG. Compared to the standard pseudo-relevance feedback approach implemented by RM1, L2EE brings significant improvements of up to 29.46% in terms of MAP, 46.52% in terms of P@10, and 14.62% in terms of nDCG. Lastly, L2EE also significantly outperforms QD, with gains of up to 23.32% in terms of MAP, 19.49% in terms

[3] A complementary evaluation of entity-oriented pseudo-relevance feedback approaches focusing on queries with named entities is conducted in the next section.

of P@10, and 7.86% in terms of nDCG. Recalling our first research question, these observations attest the effectiveness of our learning to rank approach for entity-oriented query expansion.

Note that the nDCG gain is smaller in all test collections. This occurs because different evaluation metrics have different sensitivity to ranking swaps. MAP and P@10 are based on binary relevance judgments and this might induce a different behavior compared to the nDCG scores, which are based on graded judgments [Radlinski and Craswell, 2010]. Also the relatively large gain in P@10 for CW09. Given its substantially larger size compared to the GOV2 and WT10g collections, as well as the more ambiguous nature of its associated query sets, the CW09 collection represents an arguably more challenging retrieval environment, in which the vocabulary gap between queries and documents is more pronounced. The relatively larger gains observed for CW09 suggest that query expansion can play a more noticeable role in this case, as a technique essentially aimed at improving the representation of the users' queries.

We also extend the evaluation of our supervised entity-oriented query expansion approach by comparing it with our unsupervised approach. To this end, Table 5.4 shows the retrieval performance of L2EE compared to UQEE.

Table 5.4. L2EE x UQEE retrieval performance (all queries).

WT10g		
	KL+UQEE	KL+L2EE
MAP	0.2092	**0.2628 (+25.62%)** ▲
P@10	0.3000	**0.3740 (+19.78%)** ▲
nDCG	0.4787	**0.5311 (+10.94%)** ▲
GOV2		
	KL+UQEE	KL+L2EE
MAP	0.3140	**0.3661 (+16.59%)** ▲
P@10	0.5776	**0.6866 (+18.87%)** ▲
nDCG	0.6045	**0.6418 (+6.17%)** ▲
CW09B		
	KL+UQEE	KL+L2EE
MAP	0.1633	**0.1820 (+11.45%)** ▲
P@10	0.3649	**0.3663 (+0.38%)** •
nDCG	0.3907	**0.4132 (+5.75%)** ▲

In particular, we use the most effective instance of UQEE for $k = 50$ terms, considering the experiments described in Section 3.5. This instance corresponds to UQEE-IP with the MI feature. Note that, for a fair comparison, we build both L2EE and UQEE on top of the initial baseline ranking produced by the KL retrieval model. For L2EE, percentage improvement figures compared to UQEE are shown. In addi-

tion, the aforementioned significance symbols denotes whether these improvements are statistically significant. The best value in each row is highlighted in bold.

From Table 5.4, we first observe that L2EE outperforms UQEE, with gains of up 25.62% in terms of MAP, 19.78% in terms of P@10, and 10.94% in terms of nDCG. In addition, comparing the results with Table 5.3, we observe that UQEE presents intermediate results in retrieval performance between a classical pseudo-relevance feedback approach to query expansion (RM1), and a strong entity-oriented pseudo-relevance feedback baseline (QD). Again, these observations attest the effectiveness of our supervised entity-oriented query expansion approach. Furthermore, they show that our unsupervised approach is competitive, considering the state-of-the-art pseudo-relevance feedback, and entity-oriented pseudo-relevance feedback approaches.

Effectiveness for Entity Queries

In this section, we address our second research question, by evaluating all entity-oriented pseudo-relevance feedback approaches for queries with named entities. In particular, Table 5.5 shows the retrieval performance of L2EE compared to the KL and QD baselines, considering only queries with entities. As a reference performance, we also include the oracle expansion mechanism, described in Section 5.4.1. For all entity-oriented pseudo-relevance feedback approaches, a first significance symbol denotes a statistically significant difference (or lack thereof) compared to the KL baseline. For L2EE and the oracle, a second symbol denotes significance with respect to QD. Finally, a third symbol for the oracle denotes significance compared to L2EE. The best value between baselines and L2EE is highlighted in bold.

Table 5.5. Retrieval performance (entity queries).

WT10g				
	KL	+QD	+L2EE	Oracle
MAP	0.1953	0.2320 (+18.79%) ▲	**0.3138 (+60.67%)** ▲▲	0.4295 ▲▲▲
P@10	0.2730	0.3407 (+24.80%) ▲	**0.4444 (+62.78%)** ▲▲	0.5815 ▲▲▲
nDCG	0.4686	0.5153 (+9.96%) ▲	**0.5818 (+24.15%)** ▲▲	0.6835 ▲▲▲
GOV2				
	KL	+QD	+L2EE	Oracle
MAP	0.2947	0.3131 (+6.24%) ▲	**0.3849 (+30.60%)** ▲▲	0.4362 ▲▲▲
P@10	0.5416	0.5839 (+7.81%) ▲	**0.7322 (+35.19%)** ▲▲	0.8218 ▲▲▲
nDCG	0.5860	0.6070 (+3.58%) ▲	**0.6580 (+12.29%)** ▲▲	0.7062 ▲▲▲
CW09B				
	KL	+QD	+L2EE	Oracle
MAP	0.1416	0.2050 (+44.77%) ▲	**0.2518 (+77.82%)** ▲▲	0.3044 ▲▲▲
P@10	0.2408	0.4567 (+89.66%) ▲	**0.6300 (+161.63%)** ▲▲	0.7500 ▲▲▲
nDCG	0.3720	0.4282 (+15.11%) ▲	**0.4778 (+28.44%)** ▲▲	0.5236 ▲▲▲

From Table 5.5, we observe that L2EE significantly improves upon the state-of-the-art QD baseline, with gains of up to 35.26% in terms of MAP, 37.95% in terms of P@10, and 12.90% in terms of nDCG. Recalling our second research question, these observations attest the effectiveness of our supervised approach for exploiting entity-related information for query expansion. Indeed, the improvements compared to QD are larger than those observed in the previous section, when queries that did not contain entities were also considered. On the other hand, compared to the performance of the oracle, we observe that there is still a considerable room for further improvements. For instance, considering the CW09B collection, the oracle is ahead by 20.89% in terms of MAP, 19.05% in terms of P@10, and 9.59% in terms of nDCG. In Chapter 6, we propose further directions to close this gap.

The reason why L2EE outperforms the state-of-the-art QD baseline lies in the fact L2EE is able to select terms that individually contribute more to the effectiveness of search. As an example, consider the query "poker tournaments". Table 5.6 presents the top-5 expansion terms selected by QD and L2EE considering the greater individual contribution, in terms of MAP, and the weights used for query expansion. From Table 5.6, we observe that L2EE selects terms with greater individual contribution, in terms of MAP, to the effectiveness of search. Furthermore, our approach effectively weighs the expansion terms.

Table 5.6. Top-5 expansion terms selected by QD and L2EE considering the greater individual contribution, in terms of MAP, and the weights used in expansion for the query "poker tournaments".

QD			L2EE		
Term	MAP	Weight	Term	MAP	Weight
prize	0.0553	-	world	0.0578	0.0668
limit	0.0347	-	season	0.0452	0.0298
player	0.0287	-	player	0.0287	0.0100
rebui	0.0106	-	tour	0.0287	0.0085
tabl	0.0056	-	tabl	0.0056	0.0052

Effectiveness for Difficult Queries

As discussed in Section 5.4.1, query expansion can harm the retrieval performance for difficult queries. In this section, we address our third research question, by performing a breakdown analysis of our approach according to query difficulty. To this end, we consider as difficult queries those that cannot be improved by more than 10% (in terms of MAP) using our oracle expansion mechanism. All other queries are regarded as easy. As a result of this simple quantitative criterion, we have 7 difficult queries for WT10g

(7.00%), 29 difficult queries for GOV2 (19.46%), and 11 difficult queries for CW09B (11.22%). Tables 5.7 and 5.8 show the retrieval performance of our approach compared to KL, RM1, and QD, considering difficult and easy queries, respectively. For both tables, significance symbols are defined as in Table 5.3.

Table 5.7. Retrieval performance (difficult queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.4244	0.4113 (-3.08%) $^{\bullet}$	**0.4367 (+2.89%)** $^{\bullet\bullet}$	0.4048 (-4.61%) $^{\bullet\bullet\blacktriangledown}$
P@10	0.5286	0.5000 (-5.41%) $^{\bullet}$	**0.5571 (+5.39%)** $^{\bullet\blacktriangle}$	0.5429 (+2.70%) $^{\bullet\blacktriangle\bullet}$
nDCG	0.7012	0.6914 (-1.39%) $^{\bullet}$	**0.7038 (+0.37%)** $^{\bullet\bullet}$	0.6601 (-5.86%) $^{\bullet\bullet\blacktriangledown}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.4569	**0.4686 (+2.56%)** $^{\bullet}$	0.4505 (-1.40%) $^{\bullet\bullet}$	0.4476 (-2.03%) $^{\bullet\blacktriangledown\bullet}$
P@10	0.7793	**0.8034 (+3.09%)** $^{\bullet}$	0.7690 (-1.32%) $^{\bullet\bullet}$	0.7862 (+0.88%) $^{\bullet\bullet\bullet}$
nDCG	0.7120	**0.7179 (+0.82%)** $^{\bullet}$	0.7076 (-0.61%) $^{\bullet\bullet}$	0.7070 (-0.70%) $^{\bullet\blacktriangledown\bullet}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.3221	0.3222 (+0.03%) $^{\bullet}$	0.3192 (-0.90%) $^{\bullet\bullet}$	**0.3255 (+1.05%)** $^{\bullet\bullet\bullet}$
P@10	0.5617	0.5375 (-4.31%) $^{\bullet}$	0.5590 (-0.48%) $^{\bullet\bullet}$	**0.5722 (+1.87%)** $^{\bullet\bullet\bullet}$
nDCG	0.6069	0.6063 (-0.09%) $^{\bullet}$	**0.6104 (+0.57%)** $^{\bullet\bullet}$	0.6103 (+0.56%) $^{\bullet\bullet\bullet}$

Table 5.8. Retrieval performance (easy queries).

WT10g				
	KL	+RM1	+QD	+L2EE
MAP	0.1780	0.1874 (+5.28%) $^{\bullet}$	0.1963 (+10.28%) $^{\bullet\bullet}$	**0.2521 (+41.62%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2538	0.2677 (+8.00%) $^{\bullet}$	0.2946 (+16.00%) $^{\bullet\bullet}$	**0.3613 (+44.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.4511	0.4526 (+0.33%) $^{\bullet}$	0.4765 (+5.63%) $^{\bullet\bullet}$	**0.5214 (+15.58%)** $^{\blacktriangle\blacktriangle\bullet}$
GOV2				
	KL	+RM1	+QD	+L2EE
MAP	0.2554	0.2822 (+10.49%) $^{\bullet}$	0.2934 (+14.87%) $^{\bullet\bullet}$	**0.3464 (+35.63%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.4842	0.5042 (+4.16%) $^{\bullet}$	0.5650 (+18.75%) $^{\bullet\bullet}$	**0.6625 (+37.50%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.5543	0.5733 (+3.42%) $^{\bullet}$	0.5904 (+6.51%) $^{\bullet\bullet}$	**0.6260 (+12.93%)** $^{\blacktriangle\blacktriangle\bullet}$
CW09B				
	KL	+RM1	+QD	+L2EE
MAP	0.1167	0.1181 (+1.20%) $^{\bullet}$	0.1435 (+22.96%) $^{\bullet\bullet}$	**0.1622 (+38.98%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.1966	0.2103 (+5.00%) $^{\bullet}$	0.2736 (+35.00%) $^{\blacktriangle\blacktriangle}$	**0.3379 (+70.00%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3396	0.3265 (-3.85%) $^{\bullet}$	0.3654 (+7.59%) $^{\bullet\bullet}$	**0.3860 (+13.66%)** $^{\bullet\blacktriangle\bullet}$

From Table 5.7, we observe that, although the results vary across the three considered test collections, none of the deployed query expansion approaches can significantly outperform the KL baseline. On the other hand, eventual performance drops are not significant either. For easy queries, as shown in Table 5.8, L2EE outperforms the other baselines in all cases, often significantly, with gains of up to 13.03% in terms of MAP, 15.78% in terms of P@10, and 5.63% in terms of nDCG over the second most

effective approach for each of these metrics. Recalling our third research question, the observations in Tables 5.7 and 5.8 attest the robustness of our approach in light of difficult queries, as well as its superior performance for easy queries, which comprise the majority of the query sets considered in our investigation.

Finding Non-Wikipedia Pages

The previous sections have demonstrated the effectiveness of our approach at exploiting evidence from Wikipedia in order to expand queries with named entities. A natural question that arises in this scenario is whether the observed effectiveness is merely due to an improved ability to rank Wikipedia pages themselves. In order to assess the effectiveness of our approach at ranking non-Wikipedia pages, in this section, we address our fourth research question. To this end, Table 5.9 shows the retrieval performance of our approach compared to KL, RM1, and QD, considering a modified instance of the CW09B collection without Wikipedia pages, called CW09BNW. Once again, for all query expansion approaches, we present percentage improvements over the KL baseline, with significance symbols defined as in Table 5.3.

Table 5.9. Retrieval performance on CW09BNW.

CW09BNW				
	KL	+RM1	+QD	+L2EE
MAP	0.1055	0.1089 (+3.22%) $^{\bullet}$	0.1029 (-2.46%) $^{\bullet\blacktriangledown}$	**0.1210 (+14.69%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
P@10	0.2122	0.2184 (+2.92%) $^{\bullet}$	0.1796 (-15.36%) $^{\blacktriangledown\blacktriangledown}$	**0.2562 (+20.73%)** $^{\blacktriangle\blacktriangle\blacktriangle}$
nDCG	0.3205	0.3175 (-0.94%) $^{\bullet}$	0.3216 (+0.34%) $^{\bullet\bullet}$	**0.3377 (+5.36%)** $^{\blacktriangle\blacktriangle\blacktriangle}$

Comparing results from Table 5.3 and Table 5.9, we can observe that Wikipedia pages play an important role in retrieval performance, even for the KL baseline, which performs no expansion. When we consider Wikipedia pages, the retrieval performance in terms of MAP increases 34.21% for KL, 31.22% for RM1, 60.15% for QD and 50.14% for L2EE. Thus, the existence of Wikipedia pages in the collection influences retrieval performance for all considered approaches. Moreover, it is critical for the QD baseline, which promotes the Wikipedia page related to the entity to the top of the ranking using it as the only feedback document. Additionally, L2EE is less dependent on the existence of Wikipedia pages than the QD baseline. The gain in retrieval performance provided by our method compared to the KL baseline is lower (14.69% against 28.53%), but is still significant.

From Table 5.9, we once again observe that L2EE significantly outperforms all baselines, with gains of 11.11% in terms of MAP, 17.31% in terms of P@10, and 5.01% in terms of nDCG over the second most effective approach for each metric. Recalling

our fourth research question, these observations attest the effectiveness of our entity-oriented query expansion approach to expand queries even when no Wikipedia pages are considered in the search results.

Feature Effectiveness

Throughout Section 5.4.2, we have demonstrated the effectiveness of L2EE in different scenarios, in contrast to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches. In order to further our understanding of the reasons behind such an effective performance, in this section, we address our fifth and last research question, by assessing the effectiveness of the features used by our learning to rank approach, as described in Section 5.3.2. To this end, Table 5.10 shows the retrieval performance of each of these features, when deployed in isolation in order to select the top k expansion terms for each query. As a baseline for this investigation, we include the performance of the KL retrieval model, which performs no expansion. In particular, a significance symbol denotes whether the performance of each of our considered features differs significantly from that of KL. The best value in each column is highlighted in bold.

Table 5.10. Feature retrieval performance

WT10g			
	MAP	P@10	nDCG
KL	0.1953	0.2730	0.4686
+DC	0.1882 (-3.63%) ▼	0.2870 (+5.13%) ▲	0.4574 (-2.39%) ▼
+MI	0.1875 (-3.99%) ▼	0.2900 (+6.23%) ▲	0.4560 (-2.69%) ▼
+TF	**0.2309 (+18.23%)** ▲	**0.3520 (+28.94%)** ▲	**0.5226 (+11.52%)** ▲
+TS	0.2235 (+14.44%) ▲	0.3380 (+23.81%) ▲	0.5062 (+8.02%) ▲
+TP	0.2082 (+6.60%) ▲	0.3050 (+11.72%) ▲	0.4789 (+2.20%) ▲
GOV2			
	MAP	P@10	nDCG
KL	0.2947	0.5416	0.5860
+DC	0.2898 (-1.66%) •	0.5490 (+1.37%) •	0.5780 (-1.36%) •
+MI	0.2780 (-5.67%) ▼	0.5221 (-3.60%) ▼	0.5623 (-4.04%) ▼
+TF	**0.3210 (+8.92%)** ▲	**0.6013 (+11.02%)** ▲	**0.6100 (+4.10%)** ▲
+TS	0.3098 (+5.12%) ▲	0.5725 (+5.70%) ▲	0.6015 (+2.64%) ▲
+TP	0.3043 (+3.26%) ▲	0.5846 (+7.94%) ▲	0.5936 (+1.30%) •
CW09B			
	MAP	P@10	nDCG
KL	0.1416	0.2408	0.3720
+DC	0.1595 (+12.64%) ▲	0.3714 (+54.24%) ▲	0.3863 (+3.84%) ▲
+MI	0.1578 (+11.44%) ▲	0.3643 (+51.29%) ▲	0.3863 (+3.84%) ▲
+TF	**0.1881 (+32.84%)** ▲	0.4092 (+69.93%) ▲	**0.4168 (+12.04%)** ▲
+TS	0.1822 (+28.67%) ▲	**0.4276 (+77.57%)** ▲	0.4139 (+11.26%) ▲
+TP	0.1691 (+19.42%) ▲	0.3816 (+58.47%) ▲	0.3965 (+6.59%) ▲

From Table 5.10, we first observe that both statistical features, term frequency (TF) and term spread (TS), as well as our term proximity feature (TP), perform effectively across all three considered collections, with significant improvements compared to the KL baseline in almost all settings (the only exception is for the TP feature on GOV2 in terms of nDCG). In addition, our taxonomic features, dice's coefficient (DC) and mutual information (MI), also show significant improvements on the larger CW09B corpus. Both DC and MI are high-precision features, as opposed to recall-oriented features, as indicated by their consistently positive improvements in terms of P@10 for all considered collections. While recall plays an important role for older ad-hoc test collections such as WT10g and GOV2, its importance is less pronounced for the larger CW09B collection, which comprises a considerable fraction of navigational (and hence precision-oriented) queries. Contrasting these features to one another, TF and TS are generally the most effective, followed by TP and the taxonomic DC and MI features. Lastly, compared to the results in Table 5.3, none of these features outperform their combination within our learning to rank approach, further attesting to its effectiveness. Recalling our fifth research question, these observations demonstrate the suitability of our devised features as descriptors of effective expansion terms.

5.5 Summary

This chapter extends Chapter 3 introducing another novel approach for the query expansion problem described in Chapter 2. Our supervised entity-oriented query expansion approach (L2EE) not only exploits named entities to select expansion terms, but it also weighs these terms proportionally to their predicted effectiveness.

Through Sections 5.1 and 5.3, we described the basic operation of L2EE focusing on the differences between it and our unsupervised entity-oriented query expansion approach. In particular, in Section 5.1, we described the on-line and off-line stages of L2EE retrieval process, and its main components, presenting the additional *learning to rank terms* off-line step. Furthermore, we discussed the computational overhead in query time of L2EE. In Section 5.2, we described the entity representation and resolution procedures, presenting additional article fields used by L2EE as a source of candidate expansion terms. In Section 5.3, we detailed the learning to rank approach adopted by L2EE to rank effective expansion terms. Additionally, we defined the statistical features used to instantiate the learning to rank approach within L2EE.

Finally, in Section 5.4, we presented the experimental methodology and evaluation procedures which made it possible to validate L2EE in contrast to state-of-the-art

pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches. Experimental results attested the effectiveness of our learning to rank approach for entity-oriented query expansion, with gains of up to 23.32% in terms of MAP, 19.49% in terms of P@10, and 7.86% in terms of nDCG over the most effective baseline for each of these metrics. Moreover, by breaking down our analysis by query difficulty, we demonstrated the robustness of our approach when applied for queries with little room for improvement. In addition, we showed that the observed improvements hold even when no Wikipedia pages are considered in the search results. Lastly, we analyzed the performance of each of our ranking features separately, showing that statistical and proximity features are particularly suitable for selecting effective expansion terms. Contrasting the performance of our approach to that attained by an oracle mechanism, which knows exactly whether to expand each individual query, we showed that there is still room for further improvements.

Recalling our thesis statement from Section 1.1, in this chapter we showed that the use of multiple sources of semantic evidence on entities, as well as the use of machine learning techniques to combine these features to rank candidate expansion terms are effective for query expansion. In the next chapter, we summarize the contributions and the conclusions made throughout the chapters of this thesis, also pointing directions for future research.

Chapter 6

Conclusions and Future Work

As the Web grows, web search becomes more complex. At the same time, search engines have become the primary gateway for finding information on the Web. The increasing rate of information production and consumption poses effectiveness challenges to web search engines [Santos, 2013]. Particularly, the leading web search engine has recently reported to be answering a total of 100 billion queries each month, and to be tracking over 30 trillion unique URLs [Cutts, 2012]. Given the size of the Web and the short length of typical web search queries [Jansen et al., 2000; Gabrilovich et al., 2009], there may be billions of pages matching a single query. As an immediate result, queries submitted to a web search engine are often misinterpreted, resulting in relevant documents never being retrieved, regardless of how sophisticated the subsequent ranking process is [Li, 2010]. In this scenario, an improved understanding of the information need underlying the user query becomes a challenging task.

In this thesis, we proposed to tackle query expansion by exploiting Wikipedia as a repository of feedback entities. In particular, we derived underexploited semantic evidence on entities as features to rank candidate terms, ultimately using them to expand queries. By associating entity-related information with queries, the chance of clearly understanding the information need underlying the user query can be improved. To this end, we introduced novel entity-oriented query expansion approaches aimed to improve search retrieval experience, by implicitly adding to queries candidate terms selected from entities related to them.

Throughout this thesis, we described and validated the proposed entity-oriented query expansion approaches in light of the current literature. In the remainder of this chapter, Sections 6.1 and 6.2 summarize our main contributions and the conclusions drawn from the previous chapters, respectively. In Section 6.3, we lay out several directions for future research, directly stemming from the results of this thesis.

6.1 Summary of Contributions

In the following, we summarize the main contributions of this thesis.

The use of underexploited semantic evidence on entities In Chapters 3 and 5, we presented previously underexploited sources of semantic evidence on entities. Previous work in the literature did not properly exploit the valuable human-refined information available in Wikipedia infoboxes as a source of candidate terms for query expansion. Particularly, in Sections 3.2 and 5.2, we showed how our proposed query expansion approaches incorporate features extracted from infoboxes. In addition, we described how infobox templates are used to derive a flat taxonomy of entities, where each entity is related to a unique single-level class. The generated taxonomy leverages term features previously proposed in the literature to deal properly with entities, ultimately improving the accuracy of such features to select effective candidate terms for query expansion.

A novel feature to rank expansion terms In Section 5.3.2, we proposed the term proximity (TP) feature to rank candidate expansion terms. In particular, TP uses the syntactic structure of a Wikipedia article to record how close candidate expansion terms are to the original query terms. By measuring the distance between query and candidate terms, considering the distribution of them across sentences within articles, it is feasible to properly select effective expansion terms.

An unsupervised entity-oriented query expansion approach In Chapter 3, we introduced UQEE, a novel unsupervised approach for entity-oriented query expansion. In contrast to previous approaches in the literature, UQEE takes advantage of the semantic structure implicitly provided by infobox templates, and leverage well-known discriminative features, adapting them to deal properly with entities, ultimately improving their accuracy to select effective expansion terms. Different from state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches, which need to process both the original and modified query, UQEE only processes the modified one, thus providing lower query latency.

An approach to automatically generate infoboxes In Chapter 4, we introduced WAVE, a novel self-supervised approach for autonomously extracting attribute-value pairs from Wikipedia articles. Different from previous approaches in the literature, WAVE takes advantage of the syntactic structure of a Wikipedia article to represent

it in a novel enriched plain text format, and uses a window based segmentation model to learn how to extract an unlimited number of non-predefined attribute-value pairs from them, in order to automatically generate infoboxes. Additionally, the unsupervised entity-oriented query expansion approach presented in Chapter 3 used WAVE to automatically generate infoboxes, which were shown to be as effective as manually generated infoboxes.

A supervised entity-oriented query expansion approach In Chapter 5, we introduced L2EE, a novel learning to rank approach for entity-oriented query expansion. Different from previous supervised approaches in the literature, L2EE tackles query expansion as a learning to rank problem, in order to directly learn an effective ranking of the candidate terms related to an entity in the query. As a result, not only does L2EE choose effective terms for expansion, but it also weights them proportionally to their predicted effectiveness. Similarly to our unsupervised approach, L2EE does not need to process the original query in order to select expansion terms. As it processes only the modified query, its query latency is also lower than that of state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches.

A thorough validation of the proposed approaches Throughout Chapters 4 to 5, we thoroughly validated our proposed approaches in contrast to effective approaches described in literature. In particular, Section 4.6 validated WAVE in contrast to a state-of-the-art baseline across representatives datasets, Section 3.5 validated UQEE in contrast to a standard retrieval model using a representative web test collection, and Section 5.4 validated L2EE in contrast to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches across multiple web test collections.

6.2 Summary of Conclusions

In this section, we summarize the main conclusions drawn from the thorough and comprehensive evaluation of our entity-oriented query expansion approaches, and each of their components throughout this thesis. In particular, these conclusions fully validate the statement of this thesis, as presented in Section 1.1.

On the effectiveness of Infoboxes In Sections 3.5.2 and 5.4.2, we demonstrated that entity-related information extracted from infoboxes is effective for query expansion. Particularly, in Section 3.5.2, we showed that the selection of terms directly

from infobox fields provides a better trade-off between retrieval performance and query latency. Additionally, we showed that deriving a flat taxonomy of entities using infobox templates leverage discriminative term features to deal properly with entities, ultimately improving the accuracy of such features to select effective terms for query expansion. Moreover, we showed that automatically generate infoboxes are as effective as manually generated infoboxes for query expansion. In Section 5.4.2, we showed that statistical features take advantage of the distribution of terms across multiple fields, in special infobox, leading to an effective selection of candidate expansion terms.

On the effectiveness of the TP feature In Section 5.4.2, we demonstrated the suitability of the novel term proximity feature as a descriptor of effective expansion terms. In particular, we showed that TP outperforms effective taxonomic features for query expansion. Moreover, we demonstrated that the performance of TP is strongly based on its ability to efficiently correlate by distance candidate expansion terms selected from entities in queries and query terms.

On the effectiveness of UQEE In Section 3.5, we contrasted UQEE to a standard retrieval model using a representative web test collection. The results of this investigation showed that UQEE compares favourably to the baseline approach, with significant gains in all instances. Particularly, in Section 3.5.2, we showed that all considered instances of UQEE consistently outperform the baseline approach, with relatively larger gains in instances using infoboxes, considering a fixed number of expansion terms. Additionally, we demonstrated the robustness of UQEE to balance retrieval performance and query latency. Moreover, we demonstrated the suitability of our discriminative taxonomic features to select effective expansion terms. Lastly, we analyzed the performance of automatically generated infoboxes for query expansion. Based on our observations, we attested the effectiveness of the WAVE approach, showing that it is particularly suitable to generate infoboxes with effective expansion terms. Moreover, the performance of the WAVE generated infoboxes is comparable to the manually generated ones.

On the effectiveness of WAVE In Section 4.6, we contrasted WAVE to a state-of-the-art baseline, across manually generated datasets. Note that, the four datasets were created specially for our experiments. The results of this investigation showed that WAVE compares favourably to the baseline approach, with significant gains. Particularly, in Section 4.6.2, we showed that WAVE consistently outperforms the baseline approach, with relatively larger gains in datasets with less numeric attributes. Addi-

tionally, we showed that the performance of the WAVE extractor component particularly depends on the word patterns, more regular in string attributes. Moreover, we showed that despite the gains for numeric attributes having been smaller, they are still significant, since the results of the baseline are already high, with almost no room for improvement.

On the effectiveness of L2EE In Section 5.4, we contrasted L2EE to state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches in the literature, as well as to UQEE. The results of this investigation showed that L2EE compares favourably to these approaches, with significant gains. In particular, in Section 5.4.2, we showed that L2EE consistently outperforms the considered approaches, with relatively larger gains in P@10 on the largest collection. Given the more ambiguous nature of the associated query sets for the largest collection, where the vocabulary gap between queries and documents is more pronounced, experimental results suggest that supervised entity-oriented query expansion can play a more noticeable role in this case, as a technique essentially aimed at improving the representation of the users queries. Additionally, we demonstrated the robustness of L2EE when applied for queries with little room for improvement. Moreover, we showed that the observed improvements hold even when no Wikipedia pages are considered in the search results. Lastly, we analysed the performance of each of our ranking features separately, and we observed that statistical and proximity features are particularly suitable to select effective expansion terms.

6.3 Directions for Future Research

In this section, we discuss possible directions for future research, directly inspired by or stemming from the results of this thesis. These directions cover topics related to query expansion and automatic infobox generation.

Query difficulty prediction As shown in Chapter 5, there are some queries that can not be significantly improved by any query expansion mechanism, because there is no relevant document in the collection that properly answers the query. For these difficult queries, different strategies should be considered in order to avoid an inappropriate expansion. Thus, the prediction of query difficulty is paramount to effectively address the problem of when to expand difficult queries. In this vein, we intend to exploit metrics for query performance prediction to deploy a selective query expansion mechanism, which suggests when a query should be expanded. The idea is to

extend our entity-oriented query expansion approaches by incorporating this selective mechanism. In particular, preliminary experiments using the well-known clarity score metric [Cronen-Townsend et al., 2002] showed promising results.

Machine learning Throughout this thesis, we have used machine learning to improve the estimation of several components of our approaches. For instance, in Chapter 4, we used conditional random fields for an improved estimation of the values for an article attribute, while in Chapter 5 learning to rank was used to predict effective expansion terms. Despite having been shown to be effective at estimating the various components of our approaches, the investigation of different families of machine learning algorithms can improve our understanding on the automatic infobox generation and query expansion tasks. We are currently investigating an extension of WAVE and L2EE which incorporates different families of machine learning algorithms, particularly lazy learning algorithms [Veloso et al., 2011; de Oliveira et al., 2013].

Ranking features In Chapters 3 and 5, we showed several taxonomic and information theoretic features we adapted from the literature to properly discriminate entity terms in a query expansion environment. Furthermore, we proposed a novel proximity feature for the same purpose. Experimental results attested the effectiveness of such features to rank candidate expansion terms. Nevertheless, as shown in Sections 3.5.2 and 5.4.2, the features differ significantly regarding their performance, and combining these features is an effective strategy to improve retrieval results. Thus, the investigation of other discriminative features in the literature for entity-oriented query expansion is a challenging problem. In this vein, we intend to exploit novel positional and proximity features, as well as novel statistical features in order to select effective expansion terms. Particularly, we are interested in the recently proposed maximal information coefficient (MIC) [Reshef et al., 2011], which has shown outstanding performance in different research fields.

Multiple knowledge bases Throughout this thesis, we have used Wikipedia as a source of entities from where we select effective expansion terms. Despite having been shown to be effective at providing effective expansion terms for our entity-oriented query expansion approaches, Wikipedia is only one of the high-quality knowledge bases that can be used for query expansion. Actually, previous research in the literature already consider Wikipedia [He and Ounis, 2007; Li et al., 2007; Milne et al., 2007], and other external resources, such as query logs [Cui et al., 2002], social annotation collections [Lin et al., 2011], and the ConceptNet [Kotov and Zhai, 2012] for the same task.

Recently, a combination of multiple resources was also considered [Bendersky et al., 2012; Weerkamp et al., 2012] for query expansion. We are currently investigating the use of multiple knowledge bases, other than Wikipedia, as sources of complementary information to enrich our entity repository.

Automatic generation of Infobox templates Chapter 4 introduced WAVE, an effective approach to autonomously extract attribute-value pairs from Wikipedia articles. We showed that WAVE can effectively generate infoboxes, by learning how to extract values for attributes of known infobox templates related to articles. However, rarely a Wikipedia publisher provides the infobox template for a novel article, which imposes a limitation to apply WAVE in many practical cases. Thus, the investigation of inference mechanisms able to suggest the class and attributes for a novel article based on its textual content becomes paramount to overcome this limitation. In particular, we are currently investigating information extraction methods previously proposed in the literature [Shinyama and Sekine, 2006; Paşca et al., 2007] in order to extend WAVE to properly address the infobox template generation problem.

Infobox generation difficulty prediction In some cases, it is hard to effectively generate infobox for Wikipedia articles, either because there are no infobox templates which can be properly related to the article, or the textual content of the article is insufficient to generate high quality infoboxes. In these cases, different strategies should be considered in order to avoid an inappropriate infobox generation. Thus, the prediction of infobox generation difficulty is paramount to effectively address the problem of when to generate infoboxes for articles. In this vein, we intend to exploit quality predictors for articles to deploy a selective infobox generation mechanism, which suggests when an infobox should be generated for an article. The idea is to extend WAVE by incorporating this selective mechanism.

Exploiting entities in other contexts Throughout this thesis, we shown that entities are useful for query expansion. Particularly, in Chapters 3 and 5, we presented novel query expansion approaches which use entity-related information to select effective expansion terms. Our observations suggest that entities can be useful in other contexts. Thus, the effective use of entities in multiple information retrieval scenarios becomes a challenging problem. In particular, we intend to exploit entities in other query understanding operations such as acronym expansion, query segmentation, query disambiguation, and query segmentation, as well as in other environments such as recommendation systems, e-commerce, and mobile search.

6.4 Final Remarks

This thesis presented novel approaches to address the query expansion problem. As demonstrated throughout the thesis, the principles underlying our entity-oriented query expansion approaches are general, sound, and effective. From a research perspective, the generality of the proposed approaches enabled the investigation of several dimensions of the query expansion problem, including how to exploit external knowledge bases as a source of valuable information for query expansion, how to best recognize and represent entities underlying a query, and how to estimate the predicted effectiveness of candidate expansion terms.

These investigations led to the publication of four peer-reviewed research papers directly related to this thesis. Furthermore, two other research papers, whose results indirectly corroborate this thesis, were published. Particularly, a Wikipedia processor mechanism which extracts categorical information from Wikipedia dump files was proposed by Couto et al. [2009], in order to validate the effectiveness of their classification algorithms. This is a pioneering mechanism which served as the basis for the development of the UQEE and L2EE components that build our entity repository. In addition, the near-optimal data structure proposed by Botelho et al. [2011] was used to efficiently lookup the entity repository.

Moreover, as discussed in Section 6.3, this thesis opens up directions for other researchers intending to deploy and extend our entity-oriented query expansion approaches. From a practical perspective, L2EE outperforms previous state-of-the-art pseudo-relevance feedback and entity-oriented pseudo-relevance feedback approaches. Thus, we believe that L2EE has secured its place in the state-of-the-art for entity-oriented query expansion.

Bibliography

Amati, G., Carpineto, C., and Romano, G. (2004). Query difficulty, robustness and selective application of query expansion. In *Proceedings of the 26th European Conference on Information Retrieval*, pages 127--137.

Attar, R. and Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *Journal of the ACM*, 24(3):397--417.

Auer, S. and Lehmann, J. (2007). What have Innsbruck and Leipzig in common? Extracting semantics from Wiki content. In *Proceedings of the 4th European Conference on the Semantic Web*, pages 503--517.

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern information retrieval: the concepts and technology behind search*. Pearson Education, Harlow, England, 2nd edition.

Balasubramanian, N., Kumaran, G., and Carvalho, V. R. (2010). Exploring reductions for long web queries. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 571--578.

Banerjee, S., Ramanathan, K., and Gupta, A. (2007). Clustering short texts using Wikipedia. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 787--788.

Bar-Yossef, Z. and Gurevich, M. (2008). Random sampling from a search engine's index. *Journal of the ACM*, 55(5):24:1--24:74.

Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A., and Kolcz, A. (2005). Automatic Web query classification using labeled and unlabeled training data. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 581--582.

Bendersky, M., Metzler, D., and Croft, W. B. (2012). Effective query formulation with multiple information sources. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pages 443--452.

Bergsma, S. and Wang, Q. I. (2007). Learning noun phrase query segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing*, pages 819--826.

Berners-Lee, T. (1989). Information management: A proposal. Technical report, CERN.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144--152.

Botelho, F. C., Brandão, W. C., and Ziviani, N. (2011). Minimal perfect hashing and bloom filters made practical. In *Proceedings of the IADIS International Conference Applied Computing*, pages 465--470.

Brandão, W. C., da Silva, A. S., de Moura, E. S., and Ziviani, N. (2011). Exploiting entity semantics for query expansion. In *Proceedings of the IADIS International Conference WWW/INTERNET*, pages 365--372.

Brandão, W. C., de Moura, E. S., da Silva, A. S., and Ziviani, N. (2010). A self-supervised approach for extraction of attribute-value pairs from Wikipedia articles. In *Proceedings of the 17th Symposium on String Processing and Information Retrieval*, pages 279--289.

Brandão, W. C., de Moura, E. S., Santos, R. L. T., da Silva, A. S., and Ziviani, N. (2013). Learning to Expand Queries Using Entities. *Journal of the American Society for Information Science and Technology*. Forthcoming.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the Web. In *Proceedings of the 9th International Conference on World Wide Web*, pages 309--320.

Büttcher, S., Clarke, C. L. A., and Soboroff, I. (2006). The TREC 2006 Terabyte track. In *Proceedings of 15th Text Retrieval Conference*, pages 128--142.

Cao, G., Nie, J.-Y., Gao, J., and Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM*

SIGIR Conference on Research and Development on Information Retrieval, pages 243--250.

Cao, H., Hu, D. H., Shen, D., Jiang, D., Sun, J.-T., Chen, E., and Yang, Q. (2009). Context-aware query classification. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3--10.

Carpineto, C., De Mori, R., Romano, G., and Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1--27.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1--27:27.

Chang, C.-H., Kayed, M., Girgis, M. R., and Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411--1428.

Chang, K. C.-C., He, B., Li, C., Patel, M., and Zhang, Z. (2004). Structured databases on the Web: observations and implications. *ACM SIGMOD Record*, 33(3):61--70.

Chieu, H. L. and Ng, H. T. (2003). Named entity recognition with a maximum entropy approach. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 160--163.

Chikita (2013). Chitika insights: The value of Google results positioning. Technical report, Chitika Inc., Westborough, MA, USA.

Clarke, C. L., Craswell, N., and Soboroff, I. (2009). Overview of the TREC 2009 Web track. In *Proceedings of 18th Text Retrieval Conference*.

Cortes, C. and Vapnik, V. (1995). Support-vector network. *Machine Learning*, 20(3):273--297.

Couto, T., Ziviani, N., Calado, P., Cristo, M., Gonçalves, M. A., de Moura, E. S., and Brandão, W. C. (2009). Classifying documents with link-based bibliometric measures. *Information Retrieval*, 13(4):315--345.

Croft, W. B., Bendersky, M., Li, H., and Xu, G. (2011). Query representation and understanding workshop. *SIGIR Forum*, 44(2):48--53.

Croft, W. B., Metzeler, D., and Strohman, T. (2009). *Search engines: Information retrieval in practice*. Addison-Wesley Publishing Company, 1st edition.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 299--306.

Cucerzan, S. and 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*, pages 293--300.

Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th International Conference on World Wide Web*, pages 325--332.

Cutts, M. (2012). Spotlight keynote. In *Proceedings of Search Engines Strategies*, San Francisco, CA, USA.

Dalip, D. H., Gonçalves, M. A., Cristo, M., and Calado, P. (2009). Automatic quality assessment of content created collaboratively by Web communities: A case study of Wikipedia. In *Proceedings of the 9th Joint Conference on Digital Libraries*, pages 295--304.

de Oliveira, D. M., Laender, A. H. F., Veloso, A., and da Silva, A. S. (2013). Fs-ner: a lightweight filter-stream approach to named entity recognition on twitter data. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 597--604.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 148--155.

Fernandes, D., de Moura, E. S., Ribeiro-Neto, B., da Silva, A. S., and Gonçalves, M. A. (2007). Computing block importance for searching on web sites. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 165--174.

Fetterly, D., Manasse, M., and Najork, M. (2003a). On the evolution of clusters of near-duplicate web pages. *Journal of Web Engineering*, 2(4):228--246.

Fetterly, D., Manasse, M., Najork, M., and Wiener, J. (2003b). A large-scale study of the evolution of Web pages. In *Proceedings of the 12th International Conference on World Wide Web*, pages 669--678.

Gabrilovich, E., Broder, A., Fontoura, M., Joshi, A., Josifovski, V., Riedel, L., and Zhang, T. (2009). Classifying search queries using the Web as a source of knowledge. *ACM Transactions on the Web*, 3(2):1--28.

Ganguly, D., Leveling, J., Magdy, W., and Jones, G. J. (2011). Patent query reduction using pseudo relevance feedback. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1953--1956.

Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., and Pollak, B. (2007). Towards domain-independent information extraction from web tables. In *Proceedings of the 16th International Conference on World Wide Web*, pages 71--80.

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466--471.

Gulli, A. and Signorini, A. (2005). The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th International Conference on World Wide Web*, pages 902--903.

Guo, J., Xu, G., Cheng, X., and Li, H. (2009). Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 267--274.

Habegger, B. and Quafafou, M. (2004). Building web information extraction tasks. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 349--355.

Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., and Scheel, U. (2010). Faceted wikipedia search. In *Proceedings of the 13th International Conference of Business Information Systems*, pages 1–11.

Hawking, D. and Craswell, N. (2001). Overview of the TREC 2001 Web track. In *Proceedings of 10th Text Retrieval Conference*.

He, B. and Ounis, I. (2007). Combining fields for query expansion and adaptive query expansion. *Information Processing and Management*, 43(5):1294--1307.

He, B. and Ounis, I. (2009). Finding good feedback documents. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 2011--2014.

Higashinaka, R., Dohsaka, K., and Isozaki, H. (2007). Learning to rank definitions to generate quizzes for interactive information presentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 117--120.

Hu, F., Ruan, T., Shao, Z., and Ding, J. (2011). Automatic web information extraction based on rules. In *Proceedings of the 12th International Conference on Web Information System Engineering*, pages 265--272.

Hu, M., Lim, E.-P., Sun, A., Lauw, H. W., and Vuong, B.-Q. (2007). Measuring article quality in Wikipedia: models and evaluation. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 243--252.

Jain, A., Cucerzan, S., and Azzam, S. (2007). Acronym-expansion recognition and ranking on the Web. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pages 209--214.

Jain, R. (1991). *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*. Wiley-Interscience, New York.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management*, 36(2):207--227.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of the 12th ACM Conference on Knowledge Discovery and Data Mining*, pages 217--226.

Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web*, pages 387--396.

Kaisser, M. (2008). The QuALiM question answering demo: Supplementing answers with paragraphs drawn from Wikipedia. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 32--35.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and im-

plementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881--892.

Kleinberg, J. M., Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. S. (1999). The web as a graph: measurements, models, and methods. In *Proceedings of the 5th Annual International Conference on Computing and Combinatorics*, pages 1--17.

Kotov, A. and Zhai, C. (2012). Tapping into knowledge base for concept feedback: Leveraging ConceptNet to improve search results for difficult queries. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 403--412.

Kumaran, G. and Allan, J. (2008). Effective and efficient user interaction for long queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 11--18.

Kumaran, G. and Carvalho, V. R. (2009). Reducing long queries using query quality predictors. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 564--571.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282--289.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 120--127.

Lawrence, S. and Giles, C. L. (1998). Searching the World Wide Web. *Science*, 280(5360):98--100.

Lee, C.-J., Lin, Y.-C., Chen, R.-C., and Cheng, P.-J. (2009). Selecting effective terms for query formulation. In *Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology*, pages 168--180.

Li, H. (2010). Query understanding in Web search: By large scale log data mining and statistical learning. In *Proceedings of the 2nd Workshop on NLP Challenges in the Information Explosion Era*.

Li, M., Zhu, M., Zhang, Y., and Zhou, M. (2006). Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics*, pages 1025--1032.

Li, Y., Luk, W. P. R., Ho, K. S. E., and Chung, F. L. K. (2007). Improving weak ad-hoc queries using Wikipedia as external corpus. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 797--798.

Liao, W. and Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 58--65.

Lin, Y., Lin, H., Jin, S., and Ye, Z. (2011). Social annotation in query expansion: A machine learning approach. In *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 405--414.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225--331.

Madhavan, J., Ko, D., Kot, L., Ganapathy, V., Rasmussen, A., and Halevy, A. (2008). Google's Deep Web crawl. *Proceedings of the VLDB Endowment*, 1(2):1241--1252.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press, New York, NY, USA.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 188--191.

Meadow, C. T. and Yuan, W. F. (1997). Measuring the impact of information: Defining the concepts. *Information Processing and Management*, 33(6):697--714.

Mika, P., Ciaramita, M., Zaragoza, H., and Atserias, J. (2008). Learning to tag and tagging to learn: A case study on Wikipedia. *IEEE Intelligent Systems*, 23(5):26--33.

Mikheev, A., Moens, M., and Grover, C. (1999). Named entity recognition without gazetteers. In *Proceedings of the 9th Conference on European Chapter of the Association for Computational Linguistics*, pages 1--8.

Milne, D., Witten, I. H., and Nichols, D. M. (2007). A knowledge-based search engine powered by Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 445--454.

Mitra, M., Singhal, A., and Buckley, C. (1998). Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 206--214.

Nadeau, D. and Turney, P. D. (2005). A supervised learning approach to acronym identification. In *Proceedings of the 18th International Conference on Advances in Artificial Intelligence*, pages 319--329.

Nadeau, D., Turney, P. D., and Matwin, S. (2006). Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence*, pages 266--277.

Nguyen, D. P., Matsuo, Y., and Ishizuka, M. (2007). Exploiting syntatic and semantic information for relation extraction from Wikipedia. In *Proceedings of the Workshop on Text-Mining & Link-Analysis*.

Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 61--67.

Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2013). Learning multilingual named entity recognition from wikipedia. *Journal of Artificial Intelligence*, 194:151--175.

Oliveira, V., Gomes, G., Belém, F., Brandão, W. C., Almeida, J., Ziviani, N., and Gonçalves, M. A. (2012). Automatic query expansion based on tag recommendation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1985--1989.

Paşca, M. (2007). Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management*, pages 683--690.

Paşca, M., Van Durme, B., and Garera, N. (2007). The role of documents vs. queries in extracting class attributes from text. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management*, pages 485--494.

Pant, G., Srinivasan, P., and Menczer, F. (2004). Crawling the Web. In *In Web dynamics: Adapting to change in content, size, topology and use*, pages 153--178. Springer-Verlag New York, Inc.

Park, Y. and Byrd, R. J. (2001). Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 126--133.

Peng, F., Ahmed, N., Li, X., and Lu, Y. (2007). Context sensitive stemming for Web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 639--646.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130--137.

Pôssas, B., Ziviani, N., Meira, W., and Ribeiro-Neto, B. (2005). Set-based vector model: An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems*, 23(4):397--429.

Potthast, M., Stein, B., and Anderka, M. (2008). A Wikipedia-based multilingual retrieval model. In *Proceedings of the 30th European Conference on Information Retrieval*, pages 522--530.

Radlinski, F. and Craswell, N. (2010). Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 667–674.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *Science*, 334(6062):1518--1524.

Richman, A. E. and Schone, P. (2008). Mining Wiki resources for multilingual named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 1--9.

Risvik, K. M., Mikolajewski, T., and Boros, P. (2003). Query segmentation for Web search. In *Proceedings of the 12th International Conference on World Wide Web (Posters)*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART retrieval system: Experiments in automatic document processing*, pages 313--323. Prentice Hall.

Saari, D. G. (1999). Explaining all three-alternative voting outcomes. *Journal of the American Society for Information Science*, 87(2):313--355.

Santos, R. L. T. (2013). *Explicit Web search result diversification*. PhD thesis, School of Computing Science, College of Science and Engineering, University of Glasgow.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304--311.

Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large Web search engine query log. *SIGIR Forum*, 33(1):6--12.

Spirin, N. and Han, J. (2012). Survey on web spam detection: principles and algorithms. *SIGKDD Explorations*, 13(2):50--64.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B36:111--147.

Strohman, T., Metzler, D., Turtle, H., and Croft, W. B. (2005). Indri: A language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, pages 1--6, McLean, VA, USA.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697--706.

Tan, B. and Peng, F. (2008). Unsupervised query segmentation using generative language models and Wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, pages 347--356.

Taneva, B., Cheng, T., Chakrabarti, K., and He, Y. (2013). Mining acronym expansions and their meanings using query click log. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1261--1272.

Udupa, R., Bhole, A., and Bhattacharyya, P. (2009). "A term is known by the company it keeps": On selecting a good expansion set in pseudo-relevance feedback. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 104--115.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc.

Veloso, A., Meira, W., Gonçalves, M. A., Almeida, H. M., and Zaki, M. J. (2011). Calibrated lazy associative classification. *Information Science*, 181(13):2656–2670.

Wang, P., Hu, J., Zeng, H.-J., Chen, L., and Chen, Z. (2007). Improving text classification by using encyclopedia knowledge. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 332--341.

Weerkamp, W., Balog, K., and de Rijke, M. (2012). Exploiting external collections for query expansion. *ACM Transactions on the Web*, 6(4):18:1--18:29.

Whitelaw, C., Kehlenbeck, A., Petrovic, N., and Ungar, L. (2008). Web-scale named entity recognition. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 123--132.

Witten, I. H., Moffat, A., and Bell, T. C. (1999). *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann Publishers Inc., 2nd edition.

Wu, F., Hoffmann, R., and Weld, D. S. (2008). Information extraction from Wikipedia: Moving down the long tail. In *Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, pages 731--739.

Wu, F. and Weld, D. S. (2007). Autonomously semantifying Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 41--50.

Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 4--11.

Xu, Y., Ding, F., and Wang, B. (2008). Entity-based query reformulation using Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 1441--1442.

Xu, Y., Jones, G. J., and Wang, B. (2009). Query dependent pseudo-relevance feedback based on Wikipedia. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 59--66.

Yun, B.-H. and Seo, C.-H. (2006). Information extraction from semi-structured web documents. In *Proceedings of the 1st International Conference on Knowledge Science, Engineering and Management*, pages 586--598.

Zesch, T., Gurevych, I., and Mühlhäuser, M. (2007). Analyzing and accessing Wikipedia as a lexical semantic resource. In *Proceedings of the 2007 Biannual Conference of the Society for Computational Linguistics and Language Technology*, pages 213--221.

Zhai, C. (2008). Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137--213.

Zhai, C. and Lafferty, J. D. (2001a). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 334--342.

Zhai, C. and Lafferty, J. D. (2001b). Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th ACM Conference on Information and Knowledge Management*, pages 403--410.

Zhang, W., Sim, Y. C., Su, J., and Tan, C. L. (2011). Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 1909--1914.

Zhou, G. and Su, J. (2005). Machine learning-based named entity recognition via effective integration of various evidences. *Natural Language Engineering*, 11(2):189--206.