

9-2012

Information Retrieval with Query Hypergraphs

Michael Bendersky

University of Massachusetts - Amherst, bendersky.michael@gmail.com

Follow this and additional works at: http://scholarworks.umass.edu/open_access_dissertations



Part of the [Computer Sciences Commons](#)

Recommended Citation

Bendersky, Michael, "Information Retrieval with Query Hypergraphs" (2012). *Dissertations*. 631.
http://scholarworks.umass.edu/open_access_dissertations/631

This Open Access Dissertation is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

INFORMATION RETRIEVAL WITH QUERY HYPERGRAPHS

A Dissertation Presented

by

MICHAEL BENDERSKY

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2012

Computer Science

© Copyright by Michael Bendersky 2012

All Rights Reserved

INFORMATION RETRIEVAL WITH QUERY HYPERGRAPHS

A Dissertation Presented

by

MICHAEL BENDERSKY

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

David A. Smith, Member

Rajesh Bhatt, Member

Lori A. Clarke, Department Chair
Computer Science

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, W. Bruce Croft, without whom this dissertation would not have been possible. Bruce taught me innumerable valuable lessons in conducting successful and meaningful research. His critical thinking, deep appreciation of the prior work, constant pursuit of advancing the state-of-the-art, and boundless intellectual curiosity had, and will continue to have, a profound impact on my work and my worldview.

I would also like to thank my committee members: James Allan, David Smith and Rajesh Bhatt. Their insightful comments and encouragement made this work better in many ways.

I am sincerely indebted to all the CIIR staff, past and present, for their dedicated support of my work. In particular, I would like to thank Kate Morruzzi for always having the right answer to any question, David Fisher for his help, advice and fruitful conversations over the years, and Andre Gauthier and Dan Parker for their technical expertise and support.

A special thanks goes to all the CIIR students and alumni who made the last five years a unique and unforgettable experience for me: Elif Aktolga, Niranjan Balasubramanian, Marc Cartright, Van Dang, Jeff Dalton, Fernando Diaz, Shiri Dori-Hacohen, Sam Huston, Henry Feild, Jin Young Kim, Matt Lease, Tamsin Maxwell, Hema Raghavan, Jangwon Seo, Mark Smucker, Trevor Strohman, Xiaobing Xue, Xing Yi and everyone else. I learned a great deal from my fellow CIIR students, and I will miss our passionate and fruitful conversations.

I had the good fortune to collaborate with Donald Metzler on a significant portion of this dissertation. I would like to express a special gratitude to Don for helping me

to overcome many research challenges throughout our collaboration, and his valuable insights and advice. Many aspects of this work would be incomplete without Don's help and involvement.

While at CIIR, I had the opportunity to spend a summer with Kenneth Church at Microsoft Research, and a summer with Evgeniy Gabrilovich at Yahoo! Research. These internships gave me a better appreciation of many important practical aspects of industrial research that are easy to ignore in an academic environment. I thank Kenneth and Evgeniy for these great experiences. In addition, although I did not have a chance to work with her directly, I would like to thank Susan Dumais from Microsoft Research for her valuable advice throughout my studies.

Prior to joining CIIR, I was fortunate to have Oren Kurland as my advisor at the Technion – Israel Institute of Technology. I want to thank Oren, who believed in me from the beginning and strongly supported my decision to pursue an academic career abroad.

Finally, I would like to thank the most important people in my life – my family. I thank my parents, Lora and Yakov, for teaching me the importance of a life-long commitment to learning and for their care, encouragement and unconditional love. I thank my brother, Albert, and his family, Ella, Betty and Adam, for always being there for me, in good and bad times. Finally, and most importantly, I thank Marina, my wife and love of my life, and my children, Sophie and her sibling underway. This work would not have been possible without Marina's love, patience, optimism, and advice. I am forever grateful to Marina for her unwavering support and encouragement during the last five years. To Sophie and her future sibling, I am grateful for the joy, love and wonder that they bring, and will bring, to our lives.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant IIS-0534383, in part by the Defense Advance Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part

by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the sponsor.

ABSTRACT

INFORMATION RETRIEVAL WITH QUERY HYPERGRAPHS

SEPTEMBER 2012

MICHAEL BENDERSKY

B.Sc., TECHNION, ISRAEL INSTITUTE OF TECHNOLOGY

M.Sc., TECHNION, ISRAEL INSTITUTE OF TECHNOLOGY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Current information retrieval models are optimized for retrieval with short keyword queries. In contrast, in this dissertation we focus on longer, verbose queries with more complex structure that are becoming more common in both mobile and web search. To this end, we propose an expressive query representation formalism based on query hypergraphs.

Unlike the existing query representations, query hypergraphs model the dependencies between arbitrary concepts in the query, rather than dependencies between single query terms. Query hypergraphs are parameterized by importance weights, which are assigned to concepts and concept dependencies in the query hypergraph, based on their contribution to the overall retrieval effectiveness.

Query hypergraphs are not limited to modeling the explicit query structure. Accordingly, we develop two methods for query expansion using query hypergraphs. In

these methods, the expansion concepts in the query hypergraph may come either from the retrieval corpus alone or from a combination of multiple information sources such as Wikipedia or the anchor text extracted from a large-scale web corpus.

We empirically demonstrate that query hypergraphs are consistently and significantly more effective than many of the current state-of-the-art retrieval methods, as demonstrated by the experiments on newswire and web corpora. Query hypergraphs improve the retrieval performance for all query types, and, in particular, they exhibit the highest effectiveness gains for verbose queries.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	viii
LIST OF TABLES	xiv
LIST OF FIGURES	xviii
 CHAPTER	
1. INTRODUCTION	1
1.1 The Challenge of Verbose Queries	4
1.1.1 MSN Search Query Log	5
1.1.2 Click Data Analysis	6
1.2 Complex Query Representations	8
1.3 Contributions	11
1.4 Dissertation Outline	12
2. BACKGROUND AND RELATED WORK	15
2.1 Bag-of-Words Models	15
2.2 Modeling Term Dependencies	16
2.3 Supervised Weighting in Information Retrieval	18
2.4 Indri Query Language	19
2.4.1 Concept Matching	20
2.4.2 Belief Operators	21
2.4.3 Extents	22
2.4.4 Combining Beliefs	22
2.5 Summary	23

3. QUERY HYPERGRAPHS	24
3.1 Query Representation with Hypergraphs	24
3.2 Ranking with Query Hypergraphs	28
3.3 Query Hypergraph Induction	29
3.3.1 Hypergraph Structures	29
3.3.2 Hyperedges	30
3.3.3 Factors $\phi_e(\mathbf{k}_e, D)$	32
3.3.3.1 Local Factors	33
3.3.3.2 The Global Factor	33
3.4 Query Hypergraph Parameterization	36
3.4.1 Parameterization By Structure	36
3.4.2 Parameterization By Concept	37
3.5 Parameter Optimization	40
3.5.1 Learning To Rank	40
3.5.2 Coordinate Ascent	43
3.5.3 Pipeline Optimization	44
3.6 Summary	45
4. DATASETS AND EVALUATION	46
4.1 TREC Corpora	46
4.1.1 Document Collections	47
4.1.2 Topics	47
4.1.3 Relevance Judgments	49
4.2 Evaluation	49
4.2.1 Binary Evaluation Metrics	49
4.2.2 Graded Evaluation Metrics	50
4.2.3 Statistical Significance	51
4.3 Summary	53
5. PARAMETERIZED CONCEPT WEIGHTING	54
5.1 Introduction	54
5.2 Markov Random Field for Information Retrieval	55
5.3 Weighted Sequential Dependence Model	58

5.4	Evaluation	62
5.4.1	Evaluation on TREC corpora	62
5.4.2	Evaluation on a commercial web corpus	66
5.5	Summary	67
6.	PARAMETERIZED QUERY EXPANSION	69
6.1	Introduction	69
6.2	Pseudo-Relevance Feedback	71
6.3	Parameterized Query Expansion with Query Hypergraphs	75
6.4	Parameter Optimization	78
6.5	Evaluation	81
6.5.1	Comparison with the Non-Expanded Baselines	83
6.5.2	Comparison with the Query Expansion Techniques	85
6.5.3	Robustness	88
6.6	Summary	90
7.	PARAMETERIZED QUERY EXPANSION WITH MULTIPLE INFORMATION SOURCES	93
7.1	Introduction	93
7.2	Multiple Source Expansion with Query Hypergraphs	97
7.3	Information Sources	102
7.4	Parameter Optimization	105
7.5	Evaluation	106
7.5.1	Comparison with the Non-Expanded Baselines	108
7.5.2	Comparison with the Query Expansion Techniques	109
7.5.3	Number of Expansion Terms	112
7.5.4	Impact on result diversification	113
7.5.5	Robustness	115
7.6	Summary	118
8.	PARAMETERIZED CONCEPT DEPENDENCIES	119
8.1	Introduction	119
8.2	Passages in Information Retrieval	123
8.2.1	Passage Identification	125
8.2.2	Passage-Based Retrieval Models	126

8.3	Modeling Concept Dependencies with Query Hypergraphs	127
8.4	Parameter Optimization	130
8.5	Evaluation	132
8.5.1	Comparison to the Query Likelihood Model	135
8.5.2	Comparison to the MRF-IR models	136
8.5.3	Comparison to the Weighted Sequential Dependence Model	138
8.5.4	Further Retrieval Performance Analysis	138
8.5.5	Parameterization Analysis	139
8.5.5.1	Parameterization by Structure	140
8.5.5.2	Parameterization by Concept	141
8.5.5.3	Parameterization Examples	143
8.6	Summary	143
9.	SUMMARY AND FUTURE WORK	145
9.1	Overview of the Query Hypergraphs	145
9.2	Summary of the Experimental Results	146
9.3	Future Work	148
	BIBLIOGRAPHY	151

LIST OF TABLES

Table	Page
1.1 Examples of different types of user queries, the retrieval strategies required in response to these queries and possible query formulations.	2
1.2 Summary and examples of verbose query types (spelling and punctuation of the original queries is preserved).	6
3.1 Examples of the possible structures and the concepts they might contain for a search query “members rock group nirvana”.	25
3.2 Concept importance features Φ	37
4.1 Summary of TREC document collections, topics and relevance judgments used for evaluation.	47
4.2 Graded relevance scale for the <i>ClueWeb-B</i> corpus.	48
5.1 Retrieval evaluation based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the QL and the SD methods are marked by * and †, respectively.	62
5.2 Retrieval evaluation based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the QL and the SD methods are marked by * and †, respectively.	64
5.3 Average effect of concept weighting method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the <i>MAP</i> metric).	65
5.4 Comparison of retrieval results over a sample of web queries with query likelihood (QL), sequential dependence model (SD) and the weighted sequential dependence model (WSD). Discounted cumulative gain at ranks 1 and 5 is reported.	68

6.1	Explicit and expansion concepts with the highest importance weight for the query “ <i>What is the current role of the civil air patrol and what training do participants receive?</i> ”	70
6.2	Examples of expansion terms obtained by the LCE and the PQE methods for the query “ <i>camels in north america</i> ”	79
6.3	Comparison of the parameterized query expansion method (PQE) to the non-expanded baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.	82
6.4	Comparison of the parameterized query expansion method (PQE) to the non-expanded baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and † respectively.	83
6.5	Comparison of the expansion terms obtained via pseudo-relevance feedback from the <i>Robust04</i> and the <i>ClueWeb-B</i> collections for queries “ <i>international art crime</i> ” and “ <i>dangerous vehicles</i> ”.	85
6.6	Comparison of the parameterized query expansion method (PQE) to the latent concept expansion (LCE) baseline based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method is marked by *.	86
6.7	Comparison of the parameterized query expansion method (PQE) to the latent concept expansion (LCE) baseline based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method is marked by *.	86
6.8	Comparison of the PQE method with (a) Cao et al., 2008; (b) Lv and Zhai, 2010. Best result per comparison is marked by boldface.	87
6.9	Average effect of the parameterized query expansion (PQE) method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the <i>MAP</i> metric).	90

7.1	Comparison of the performance of the latent concept expansion (LCE) with retrieval corpus or Wikipedia to the performance of the query expansion using multiple information sources (MSE) for the query “ <i>ER TV Show</i> ”.	94
7.2	External information sources used in the multiple source expansion (MSE) method.	103
7.3	Comparison between the lists of expansion terms derived from the individual external information sources for the query “ <i>toxic chemical weapon</i> ” and the combined list produced by the MSE method.	103
7.4	Comparison of the parameterized query expansion methods to the non-expanded baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.	107
7.5	Comparison of the parameterized query expansion methods to the non-expanded baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.	108
7.6	Comparison of the parameterized query expansion methods to the query expansion baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method, the LCE-WP method and the PQE methods are marked by *, †, and ‡ respectively.	109
7.7	Comparison of the parameterized query expansion methods to the query expansion baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method, the LCE-WP method and the PQE methods are marked by *, †, and ‡ respectively.	110
7.8	Result diversification performance (<i>ClueWeb-B</i>). Statistically significant difference of MSE over the baselines are marked using *, †, and ‡, for WSD, PQE and LCE-WP baselines, respectively. Best result per column is marked by boldface.	113

7.9	Average effect of the parameterized query expansion (MSE) method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the <i>MAP</i> metric).	115
8.1	Retrieval baselines and their respective query hypergraph representation including the global hyperedge. \mathcal{S} indicates parameterization by structure, \mathcal{C} indicates parameterization by concept.	132
8.2	Evaluation of the performance of the retrieval with query hypergraphs using binary metrics. Best result per column is marked in boldface. Statistically significant differences with a non-hypergraph baseline are marked by its title initial.	133
8.3	Evaluation of the performance of the retrieval with query hypergraphs using graded metrics. Best result per column is marked in boldface. Statistically significant differences with a non-hypergraph baseline are marked by its title initial.	134
8.4	Query hypergraph parameterization by structure (<i>Robust04</i> collection).	141
8.5	Query hypergraph parameterization by concept (<i>Robust04</i> collection).	141
8.6	Examples of weights assigned to the concepts in the local and global factors.	142
9.1	Retrieval effectiveness gains, as measured by <i>MAP</i> , of query hypergraph based retrieval models (WSD , \mathcal{H} - WSD) compared to the current state-of-the-art retrieval models (QL , SD). The numbers in the parentheses indicate the percentage of improvement in <i>MAP</i> over the QL baseline. Statistically significant improvements with respect to QL and SD are marked by * and †, respectively.	147
9.2	Retrieval effectiveness gains, as measured by <i>MAP</i> , of query hypergraph based retrieval models that incorporate query expansion (PQE , MSE) compared to the latent concept expansion model (LCE). The numbers in the parentheses indicate the percentage of improvement in <i>MAP</i> over the LCE baseline. Statistically significant improvements with respect to LCE is marked by *.	147

LIST OF FIGURES

Figure	Page
1.1	Boxplot of the distribution of the average click positions per query for different query types. 7
1.2	A schematic drawing of an hierarchical query representation for query containing three explicit terms A , B , and C and two expansion terms E and F . The query representation conforms to the five desiderata in Section 1.2. Circles represent query concepts. Concept weights are marked by the circle size. 11
3.1	Example of a hypergraph representation for the query “ <i>international art crime</i> ”. 25
3.2	Excerpt a relevant document retrieved in response to the query “ <i>Provide information on the use of dogs worldwide for law enforcement purposes</i> ”. Non-stopword query terms are marked in boldface. 34
3.3	The outline of the coordinate ascent optimization algorithm. 42
3.4	The outline of the pipeline optimization. 44
4.1	An example of $\langle title \rangle$ and $\langle desc \rangle$ queries in a TREC topic §53. 47
5.1	A Markov random field model for a three-term query under the sequential dependence assumption. 55
5.2	A hypergraph H^{SD} that encodes the sequential dependence model for a three-term query. 58
5.3	Examples of weighted $\langle title \rangle$ and $\langle desc \rangle$ queries for TREC topic §664. Common stopwords are automatically removed from the queries prior to weight assignment. 60
6.1	Schematic diagram of query expansion using pseudo-relevance feedback from the retrieval corpus. 72

6.2	A hypergraph H^{PQE} that encodes the parameterized query expansion model for a three-term query.	75
6.3	Pipeline optimization of the parameterized query expansion method.	79
6.4	Robustness of the LCE and PQE methods for the $\langle desc \rangle$ queries with respect to the QL method.	91
7.1	Schematic diagram of query expansion with three information sources: retrieval corpus, Wikipedia, and anchor text.	95
7.2	Two hypergraphs that encode the multiple source expansion model for a three-term query with three information sources.	98
7.3	Pipeline optimization of the multiple source expansion method.	105
7.4	Varying the number of expansion terms (<i>ClueWeb-B</i> corpus). Dotted line indicates the performance of LCE[10]. Dashed and solid lines represent the performance of LCE-WP[N] and MSF[N], respectively.	112
7.5	Robustness of the LCE and MSE methods for the $\langle desc \rangle$ queries with respect to the QL method.	116
8.1	Excerpts from (a) the top document retrieved by the sequential dependence model, and (b) the top document retrieved using a query hypergraph in response to the query: “ <i>Provide information on the use of dogs worldwide for law enforcement purposes</i> ”. Non-stopword query terms are marked in boldface.	120
8.2	Overlapping passage identification.	125
8.3	Bipartite graph representation of concept dependencies in a query hypergraph H . Local edges are represented by the solid edges in the bipartite graph. The global hyperedge is represented by the dashed edges in the bipartite graph.	128
8.4	Pipeline optimization of the parameterized query hypergraph with a global hyperedge.	130

CHAPTER 1

INTRODUCTION

Typically, queries in information retrieval applications are represented as *bags-of-words*. That is, query terms are assumed to be independent from one another. While simplistic, the bag-of-words assumption has been useful for creating many successful retrieval models in the past. However, it becomes less realistic as information retrieval becomes more integrated in applications beyond web search, and user search queries become more diverse, complex and verbose.

In web search, the most well-known information retrieval application today, users commonly use short *keyword queries* that have very simple grammatical structures. Keyword queries usually contain no more than three terms (BENDERSKY and CROFT 2009) most of which are proper nouns (BARR *et al.* 2008), and are frequently used for navigational purposes, i.e., to find a particular web page (BRODER 2002).

In contrast, *verbose queries*, which are the focus of this dissertation, are long, linguistically rich expressions of user information needs. In many cases, verbose queries are expressed as natural language questions or sentences and contain multiple parts of speech, complex grammatical structures, and redundancies. Oftentimes, verbose queries can take forms that are very different from the typical *wh*-questions. Therefore, a robust combination of diverse retrieval strategies is required to improve search with verbose search queries.

To illustrate this point, consider the different types of queries shown in Table 1.1. As can be seen in Table 1.1, the short keyword queries can be usually resolved by a URL match (query (a)) or an exact phrase match (query (b)). The longer, more

User Query	Retrieval Strategy
(a) <i>facebook</i> ⇒ <code>site:facebook.com</code>	URL match
(b) <i>old bangkok inn</i> ⇒ "old bangkok inn"	Exact phrase match
(c) <i>What should I bring when traveling to Bolivia?</i> ⇒ <code>travel to bolivia</code>	Redundancy elimination
(d) <i>the in laws with michael douglas</i> ⇒ "The In-Laws" + "Michael Douglas"	Entity detection
(e) <i>budget accommodation in Bangkok that is near the subway station</i> ⇒ "guesthouse near subway" + Bangkok	Long-range dependency detection, query expansion

Table 1.1. Examples of different types of user queries, the retrieval strategies required in response to these queries and possible query formulations.

verbose queries in Table 1.1 may require additional linguistic processing and more complex retrieval strategies such as removal of the redundant linguistic structures (query (c)), entity detection (query (d)), long-range dependency detection and query expansion using related terms (query (e)).

These complex retrieval strategies pose many interesting challenges to the standard web search engines. To address these challenges, a number of what can be broadly referred to as *semantic search engines* have gained popularity. Examples include Powerset, WolframAlpha, Hokia, DeepDyve, True Knowledge and, most recently, IBM’s DeepQA system (FERRUCCI *et al.* 2010) to name just a few. While some of these semantic search engines have been successful within a restricted domain (for instance, the DeepQA system defeating the champions of the Jeopardy TV show), an end-to-end solution that handles equally well all types of queries ranging from keyword queries to verbose natural language queries, and works on the scale of the entire web is yet to emerge.

Thus, developing effective and robust retrieval models that can handle both keyword queries and verbose queries is important both from the scientific and the com-

mercial perspectives. This is further strengthened by the fact that the share of verbose queries in search engine traffic has been growing steadily in the last few years.

For instance, according to the recent *Hitwise* press releases, the number of queries with 5+ words in the web search traffic grew by 10% in 2008¹. It grew by an additional 5%, in 2009². According to *Hitwise*³, as of September 2011, the share of 5+ word queries in the entire query traffic is 18%.

In addition, while the verbose queries may still constitute a small portion of web search query traffic, they are very common for complex informational search activities such as search in Question Answering archives, patent search, enterprise search, academic search, and legal search. They are also proving to be important for voice-activated search on mobile devices (FENG *et al.* 2011).

Finally, verbose queries are highly pertinent in the emerging social search medium, since social networks such as Facebook, Twitter, Quora and others, allow users to directly communicate their information needs to other users. The types of questions that people ask in social networks are very different to the ones used in search engines, being longer and more grammatically complex (JEON *et al.* 2005; HOROWITZ and KAMVAR 2010; HECHT *et al.* 2012). Whether it is to route the question to the most suitable person, or to find an answer to a similar question, it is important to achieve a better understanding of verbose queries used in the social search applications.

To overcome the unrealistic simplicity of the commonly used bag-of-words retrieval models, researchers started to investigate term dependencies in search queries. This led to the creation of successful retrieval models, especially for large-scale web collections (METZLER and CROFT 2005; MISHNE and DE RIJKE 2005; BAI *et al.* 2008).

¹<http://www.hitwise.com/us/press-center/press-releases/google-searches-jan-09/>

²<http://www.hitwise.com/us/press-center/press-releases/google-searches-jan-10/>

³<http://www.experian.com/hitwise/press-release-google-share-of-searches-sept-2011.html>

Motivated by these models, in this dissertation, we present a formal query representation and retrieval framework that goes beyond term dependencies. This framework facilitates modeling of more complex linguistic phenomena in verbose queries by integrating weighted evidence from multiple query representations.

Our framework represents a query as a *hypergraph of concept structures*. We demonstrate that this query representation relaxes some of the independence assumptions made in previous work, and supports the development of complex and realistic query representations. These representations can be applied to improve the retrieval effectiveness of a search engine, especially for verbose queries, which – as shown in Table 1.1 – exhibit more complex linguistic structures than short keyword queries.

As we show in this dissertation, the query hypergraph representation leads to the development of several retrieval models that incorporate concept weighting, query expansion and concept dependencies. These retrieval models are significantly more effective than the current state-of-the-art retrieval models, especially for verbose queries.

The rest of this chapter is organized as follows. In Section 1.1 we illustrate some of challenges of information retrieval with verbose queries by analyzing user behavior reflected in a commercial search engine query log. Motivated by the challenges that the verbose queries pose to information retrieval systems, in Section 1.2 we introduce the hierarchical query representation that may help in improving the effectiveness of these queries and show how this query representation can be modeled using query hypergraphs. In Section 1.3 we state the main contributions of this dissertation. Finally, in Section 1.4 we provide the outline of the remainder of the dissertation.

1.1 The Challenge of Verbose Queries

While the general query representation framework described in this dissertation is robust enough to handle multiple query types, a major motivation of this work is information retrieval with verbose queries. In this section, we explain this motiva-

tion, and analyze a commercial search engine query log to demonstrate some of the challenges that the verbose queries present to the current search engines.

1.1.1 MSN Search Query Log

For our analysis we use an excerpt from an MSN Search query log. This query log includes around 15 million queries and user click data associated with these queries, sampled over a period of one month in 2006⁴.

User activity recorded in commercial search engine logs has proved to be a valuable resource for the researchers in the fields of information retrieval, data mining, machine learning and natural language processing. Large volumes of user queries and the corresponding click data in the search logs were successfully leveraged for providing an insight into the searcher behavior (JONES and KLINKNER 2008; MEI and CHURCH 2008; DOWNEY *et al.* 2008).

In this section, we examine a relatively small yet significant segment of verbose queries in the query log. Most queries in the search log are short. Query length (measured by the number of query terms) follows a power-law distribution, with the verbose queries in the tail. In fact the average number of terms per query in the MSN query log is 2.4, and queries with less than five terms account for 90.3% of the total queries.

Due to our focus on the verbose queries in this dissertation, we divide the queries in the log into two (unequally sized) main types: *short* and *verbose*. For simplicity, the division is based on the number of terms in the query.

- Short keyword queries are queries with less than five terms.
- Verbose queries are queries with five or more terms.

⁴See <http://research.microsoft.com/en-us/um/people/nickcr/wscd09/> for more details about this dataset.

Total Queries: 14,921,286			
Verbose Queries : 1,423,664			
Type	Avg. Length	Count	% of Verbose
Composite (CO)	5.67	910,103	63.93
<i>Queries that can be composed using the short queries in the query log</i> T.I. the rapper web site merryhill schools a noble learning community			
Noun Phrases (NC_NO)	5.77	209,906	14.74
<i>Noun phrase non-composite queries</i> Hp pavilion 503n sound drive lessons about children in the bible			
Verb Phrases (NC_VE)	6.35	118,736	8.34
<i>Verb phrase non-composite queries</i> detect a leak in the pool eye hard to open upon waking in the morning			
Questions (QE)	6.75	106,587	7.49
<i>Wh-questions</i> What is the source of ozone? how to feed meat chickens to prevent leg problems			

Table 1.2. Summary and examples of verbose query types (spelling and punctuation of the original queries is preserved).

All the short queries are assigned to a type SH, while the verbose queries are divided between five mutually exclusive types, which are summarized in Table 1.2.

Verbose queries are much less frequent than the short ones and therefore have a much sparser associated click data. Click data is crucial for predicting which results will be relevant for a particular query (JOACHIMS 2002) and therefore, it is not surprising that the relevance of the results presented to the users is lower for the verbose queries when compared to the short keyword queries, as demonstrated by the click data analysis we present in the next section.

1.1.2 Click Data Analysis

The types of queries in Table 1.2 are derived from the structure of the query strings. However, although the proposed taxonomy is reasonable from a syntactical point of view, we are more interested in its utility for analyzing the quality of

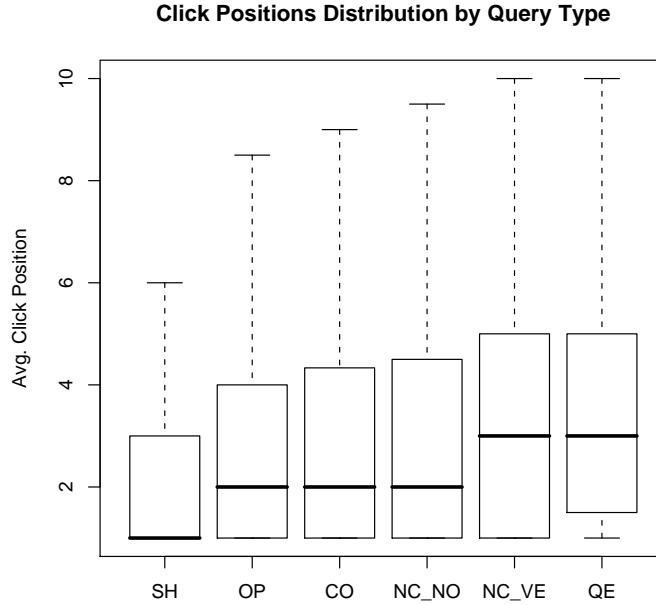


Figure 1.1. Boxplot of the distribution of the average click positions per query for different query types.

retrieval with verbose queries. Accordingly, in this section we explore whether the users interaction with the search engine differs for each of the query types.

Figure 1.1 shows the distribution of the average click position for the six types of queries (the short queries and the five types of the verbose queries) in a random sample of 10,000 queries per query type. Note that a larger value in the boxplot translates into a *lower* position of the click in the ranked list. For example, for the short queries (type **SH**) the median of the average click positions is the first result in the ranked list, while for the question queries (type **QE**), the median is the third result.

Figure 1.1 demonstrates that (a) on average, users tend to click lower in the result list for the verbose queries than for the short ones, and (b) there are differences in user click behavior between the different types of verbose queries. Specifically for

the verbose queries, operators, composite queries and noun phrases are more effective than verb phrases and questions.

Overall, as Figure 1.1 shows, capturing the linguistic structure of search queries is important for the purpose of information retrieval. The most significant drops in click position occur for the queries with complex grammatical structures such as questions and verb phrases. This demonstrates that the user click behavior is strongly dependent on the query structure.

1.2 Complex Query Representations

As the analysis in the previous section shows, there is a need to develop robust and effective query representation methods that go beyond bag-of-words and term dependencies that are commonly used in web search (METZLER and CROFT 2005; MISHNE and DE RIJKE 2005; BRIN and PAGE 1998) in order to improve the retrieval effectiveness of verbose queries. The existing simple query representations are often insufficient to accurately model the complex grammatical structure of verbose queries such as questions or verbal phrases. Therefore, in this section, we introduce an outline of a comprehensive query representation method, query hypergraphs, that is proposed in this dissertation.

To motivate the query hypergraph representation proposed in this dissertation, we describe five desiderata for verbose query representation that are based on the query analysis in the previous sections.

- (a) **Hierarchical Query Structure.** A simple method for inducing query structure is to assign each query term to a single concept. That is, a standard bag-of-words query representation is a special case of query structure. Other methods that may be used to induce structure over a query include (but are not limited to) sequential dependence modeling (a concept corresponds to a bigram) (METZLER and CROFT 2005), noun phrase chunking (a concept corre-

sponds to a noun phrase) (BENDERSKY and CROFT 2008), query segmentation (BERGSMA and WANG 2007), and dependence parsing (a concept corresponds to a sub-tree of a parse) (PARK and CROFT 2010). To integrate these multiple ways of query structure induction, an effective query representation method must support a hierarchical combination of query structures. First, we assume that we can induce a set of linguistic structures from the surface form of the query. Each of these structures can then be decomposed into atomic units, or concepts (terms, bigrams, noun phrases, etc.).

- (b) **Concept Weighting.** Some of the query concepts may be more important than others. For instance, in the query (c) in Table 1.1 (*What to bring when traveling to Bolivia?*), the verb *bring* is less important than the verb *travel*, and both of them are less important than the destination in question, *Bolivia*. Therefore, an effective query representation must support assignment of weights to individual concepts derived from the query (terms, bigrams, noun phrases, etc.). These weights should reflect the importance of the concept for retrieving the most relevant documents in response to the query.
- (c) **Query Expansion.** In some cases, the query itself does not always contain all the concepts necessary for finding all the relevant documents. For instance, in the case of query (e) in Table 1.1, adding terms such as *motel* or *guesthouse* to the original query (*budget accommodation in Bangkok that is near the subway station*) may help to retrieve more pages about budget accommodations in Bangkok. Since such query expansion with related terms or concepts is a common practice in many information retrieval applications (LAVRENKO and CROFT 2003; METZLER and CROFT 2007a; XU and CROFT 1996), an effective query representation must be flexible enough to accommodate structures that do not explicitly occur in the query.

- (d) **Concept Dependencies.** As mentioned above, term dependencies alone are not always enough to capture the linguistic richness of verbose queries. For instance, in the case of query (e) in Table 1.1, we would like to model not only the dependence between the terms *budget* and *accommodation*, but also a dependency between the phrase *budget accommodation* and the terms *Bangkok* and *subway*. Therefore, an effective query representations must support dependencies between arbitrary concepts, rather than single terms, i.e., high-order term dependencies.
- (e) **Parameter Optimization.** Since the ultimate goal of query representation is information retrieval, query representation must be an integral part of the retrieval model. In other words, the parameters that govern the query representation must also govern the retrieval model. In such a way, optimizing the parameters of the query representation will directly result in a better retrieval performance.

Figure 1.2 shows a schematic drawing of the query representation as defined by these desiderata. A set of structures is first induced over both the explicit query concepts and the expansion terms related to the query. Then, concepts in each structure are weighted based on their importance for the retrieval performance (in the case of Figure 1.2, C is the most important concept). The arcs in Figure 1.2 represent a concept dependency between the term C and the phrases AB and BC .

In the following chapters, we fully formalize this query representation and the corresponding desiderata using the *query hypergraph* representation. As we show in this dissertation, query hypergraphs can be used to instantiate a variety of query representations and retrieval models that are significantly more expressive and effective than the current state-of-the-art retrieval techniques. While the main motivation of our work is verbose queries, we show that query hypergraphs are robust enough

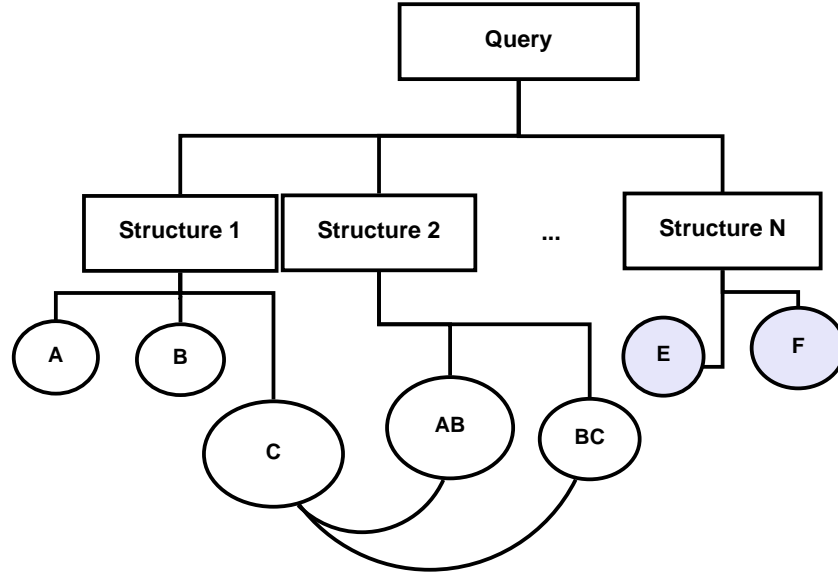


Figure 1.2. A schematic drawing of an hierarchical query representation for query containing three explicit terms A , B , and C and two expansion terms E and F . The query representation conforms to the five desiderata in Section 1.2. Circles represent query concepts. Concept weights are marked by the circle size.

to handle a variety of query types, ranging from short keyword queries to verbose natural language queries.

1.3 Contributions

In this section, we summarize the main contributions of this dissertation.

- (a) We propose a novel query representation formalism called *query hypergraphs*. Unlike the existing query representations, query hypergraphs may be used to not only model the dependencies between single query terms, but also the dependencies between arbitrary concepts in the query. Therefore, query hypergraph representation is among the first publicly available methods that model higher-order term dependencies in the query.
- (b) We propose a novel method for *query hypergraph parameterization* that enables the assignment of weights to concepts and concept dependencies in the query

hypergraph according to their contribution to the overall retrieval effectiveness of the query.

- (c) In addition to using the query hypergraphs in order to represent the explicit query structure, we also propose a simple method to model *query expansion using query hypergraphs*. The expansion concepts in the expanded query hypergraph may come from a variety of information sources, including the target retrieval corpus or an external document collection such as Wikipedia.
- (d) We propose a *pipeline optimization procedure* to estimate the parameters of the complex query hypergraphs that incorporate multiple concept dependencies or expansion concepts. Query hypergraph parameters are optimized to achieve the maximal retrieval effectiveness and thus overcome the metric divergence problem.
- (e) We empirically demonstrate the effectiveness of the proposed query hypergraphs for document retrieval. Query hypergraphs are significantly more effective (as measured by a number of standard information retrieval metrics) than any of the current state-of-the-art retrieval methods. These effectiveness gains are consistent across both newswire and web corpora.
- (f) The main focus of this dissertation is on improving the retrieval performance of verbose queries. We empirically demonstrate that, for verbose queries, query hypergraphs exhibit consistently high effectiveness gains compared to the other methods. However, query hypergraph representation is robust enough to handle both short and verbose queries, and it is significantly more effective than any of the existing retrieval methods for both query types.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows.

- (a) In Chapter 2, we survey the related work, as well as provide some information about the Indri query language, which is used to instantiate the query hypergraph representations in the experiments in this dissertation.
- (b) In Chapter 3, we provide a formal definition of query hypergraphs, their induction process and their use for document retrieval. In addition, we describe the pipeline optimization procedure to estimate the parameters of the complex query hypergraphs.
- (c) In Chapter 4, we describe the constituents of TREC corpora used for empirical evaluation of our retrieval methods. In addition, we describe the evaluation metrics used in this dissertation.
- (d) In Chapter 5, we present parameterized concept weighting, a method to assign weights to the concepts in the query based on their contribution to overall query effectiveness. In particular, we show that we can model a weighted variant of a sequential dependence model, state-of-the-art retrieval model (METZLER and CROFT 2005), using a query hypergraph representation.
- (e) In Chapter 6, we present parameterized query expansion, which goes beyond assigning weights to explicit query concepts. Parameterized query expansion allows the assignment of related concepts from the retrieval corpus to the original query, and parameterized weights to these concepts. This results in a fully weighted query representation using a hypergraph that integrates both explicit and expansion concepts.
- (f) In Chapter 7, we present multiple source expansion, which enables query expansion using multiple information sources. Multiple source expansion is especially helpful in situations when the retrieval corpus does not yield sufficiently relevant expansion concepts. Multiple source expansion results in a fully weighted query

representation using a hypergraph that integrates both explicit query concepts and expansion concepts from multiple information sources.

- (g) In Chapter 8, we present parameterized concept dependencies, a novel technique to model dependencies between arbitrary concepts in the query. Parameterized concept dependencies are also weighted by their contribution to the overall query effectiveness. These concept dependencies can be integrated in various query representations as hyperedges in a query hypergraph.
- (h) In Chapter 9, we summarize the findings of this dissertations and propose some promising directions for future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we survey the related work on bag-of-words retrieval models (Section 2.1), retrieval models that incorporate term dependencies (Section 2.2) and retrieval models that incorporate supervised term and concept weighting (Section 2.3). In addition, in Section 2.4, we introduce the Indri query language (STROHMAN *et al.* 2004), which is used in our experiments to instantiate the proposed query representations.

2.1 Bag-of-Words Models

Traditionally, formal retrieval models treat queries as bags of words. Examples of such retrieval models include (among many others): vector space model (SALTON *et al.* 1975), BIR model (ROBERTSON and SPARCK JONES 1988), BM25 model (ROBERTSON and WALKER 1994), query likelihood model (PONTE and CROFT 1998), and divergence from randomness model (AMATI and VAN RIJSBERGEN 2002).

The bag-of-words models assume that queries have a very simple linguistic structure: concepts are query terms, and there are no dependencies between the different concepts. This is a very limiting assumption, which to a large degree ignores the linguistic structure of the search query. However, until recently there was little evidence that going beyond bag of words representations consistently improves the effectiveness of the existing retrieval methods (SALTON and BUCKLEY 1988).

Term weighting plays an important role in the bag of words models. The term weighting in these models is based on either the inverse document frequency (IDF)

of the term (SALTON *et al.* 1975; ROBERTSON and WALKER 1994; SALTON and BUCKLEY 1988; ZOBEL and MOFFAT 1998), or the inverse collection frequency (ICF) of the term (ZHAI and LAFFERTY 2004; AMATI and VAN RIJSBERGEN 2002; KWOK 1990; SMUCKER and ALLAN 2006). There are many variants of term weighting schemes used in different retrieval models. For instance, ZOBEL and MOFFAT (1998) show ten examples of term weighting schemes based on IDF alone. Of these weighting schemes, “none was shown to be consistently valuable across all of the experimental domains” (ZOBEL and MOFFAT 1998).

One of the goals of this dissertation is to address the issue of query term and concept weighting in a principled manner. In this dissertation we propose a concept weight optimization based on the optimization of some retrieval metric of interest (e.g., average precision or normalized discounted cumulative gain – refer to Section 4.2 for more details on these metrics).

2.2 Modeling Term Dependencies

Recently, there has been a resurgence of interest in retrieval models that go beyond bags of words. This resurgence was mainly motivated by gains in retrieval effectiveness, which were observed on large-scale web collections, when term dependencies were incorporated into the retrieval model (METZLER and CROFT 2005; MISHNE and DE RIJKE 2005; BAI *et al.* 2008; PENG *et al.* 2007; SVORE *et al.* 2010).

Most of these term dependence models, however, take several simplifying assumptions. First, they only consider a single term dependence type (or a handful of types). For instance, METZLER and CROFT (2005) consider dependencies between adjacent query term pairs, TAO and ZHAI (2007) consider all the term pairs in the query, NALLAPATI and ALLAN (2002) consider term dependencies in a maximum spanning tree of the query (based on term co-occurrence), and GAO *et al.* (2005) consider syntactic

phrases. In contrast, we propose a retrieval model that allows combining concepts, rather than terms, and which can model various dependence types.

Second, most of these models do not explicitly assign weights to different term dependencies (METZLER and CROFT 2005; PENG *et al.* 2007; TAO and ZHAI 2007). This can be especially detrimental for models that have an exponential number of term dependencies (for instance, the full dependence model proposed by METZLER and CROFT (2005)).

Third, the majority of the term dependence models consider only first-order term dependencies. In other words, these models only consider the dependencies between the *terms*, and disregard the dependencies between the *term dependencies* (modeled as concepts in our query representation). This creates an over-simplified model of the query structure, especially for verbose natural language queries.

Query hypergraphs, which we describe in this dissertation, address the three issues above. First, they allow us to incorporate multiple concept types into the ranking function through the hierarchy of structures (see Figure 1.2). Second, they provide a principled way to weight both the structures and the concepts within the structures, such that the retrieval performance is optimized. Finally, they allow modeling higher-order dependencies between arbitrary concepts, rather than just single terms.

To the best of our knowledge, there is very little prior work on retrieval with higher-order term dependencies (i.e., dependencies between arbitrary concepts rather than terms). One notable exception is an early work on generalized term dependencies by YU *et al.* (1983), which derives higher-order dependencies from pairwise term dependencies. However, the model proposed by YU *et al.* (1983) is infeasible for large scale collections, since it requires an explicit computation of the probability of relevance for each individual query term, as well as pairs and triples of query terms.

A more recent retrieval model that attempts to incorporate higher-order term dependencies is the Full Dependence (FD) variant of the Markov random field model

proposed by METZLER and CROFT (2005). The FD model, however, is only able to capture dependencies between multiple terms, rather than multiple concepts. For instance, it can model a dependency between the terms in the triple (*dogs, law, enforcement*), but it cannot model a dependency between the pair of concepts (*dogs, “law enforcement”*).

2.3 Supervised Weighting in Information Retrieval

In the last several years, information retrieval researchers started to explore supervised models for term and concept weighting. These models facilitate more effective weighting schemes than the traditional TF-IDF weighting (SALTON *et al.* 1975), especially for more verbose queries.

BENDERSKY and CROFT (2008) treated the problem of concept weighting as a classification problem, in which noun phrase concepts in the query are labeled as either *key* or *non-key* concepts. Then, an AdaBoost classifier is trained to classify each noun phrase concept into either a *key* or a *non-key* class using a combination of statistical and syntactic features. The probability that the concept belongs to a key class is then used for concept weighting in the query. BENDERSKY and CROFT (2008) show that using as few as two weighted noun phrase concepts (in addition to the original query) can significantly improve the retrieval performance for verbose natural language queries.

Similarly to BENDERSKY and CROFT (2008), ZHAO and CALLAN (2010) use the probability of *term necessity* as a weighting mechanism for the query terms. The necessity of term t is defined as the probability of a term t occurring in documents relevant to a given query Q , i.e. $P(t|\mathcal{R})$, where \mathcal{R} is the set of relevant documents for query Q . The advantage of term necessity weighting over the key concept weighting is that it leverages the existing relevance labels, and does not require an additional

labeling of key and non-key concepts. However, it has an important disadvantage of operating on the level of single terms rather than arbitrary concepts.

To integrate the term weighting more tightly into the retrieval framework, Lease (LEASE *et al.* 2009; LEASE 2009) proposed a RegressionRank method, which utilizes expected mean average precision as a target metric. The RegressionRank weighting approach showed significant retrieval effectiveness improvements when integrated either in a bag of words model (LEASE *et al.* 2009) and a term dependency model (LEASE 2009).

CAO *et al.* (2008) extend these approaches beyond the terms that explicitly occur in the query. Their method applies term weighting to the expansion terms as well. They train a weighting model that distinguishes between *good* and *bad* expansion terms and show that their weighting scheme outperforms a standard query expansion mechanisms such as relevance models (LAVRENKO and CROFT 2003).

Query hypergraphs, which are the focus of this dissertation, present an important advance compared to these existing term and concept weighting models. First, query hypergraphs present a principled approach for weighting arbitrary concept types, rather than just a single concept type such as a term or a noun phrase. Second, they directly integrate the concept weighting into the retrieval model. Third, they are able to simultaneously optimize both explicit query concept weights and expansion concept weights. Finally, they are able to assign weight to *concept dependencies* as well as to single concepts.

2.4 Indri Query Language

The query hypergraph representation described in this dissertation can be viewed as a special case of structured query representation. Therefore, in this section, we describe the Indri query language (STROHMAN *et al.* 2004) which facilitates structural

query representation and is used to instantiate all the query hypergraph variants discussed in this dissertation.

The Indri query language and its underlying retrieval model (STROHMAN *et al.* 2004) combine the language modeling (PONTE and CROFT 1998) and the inference network (TURTLE and CROFT 1991) approaches to information retrieval. The resulting model allows rich, structured query representations to be evaluated using language modeling estimates within the inference network.

While the Indri query language is flexible enough to enable very rich query representations, it lacks a formal mechanism for automatically converting a given keyword query into its structured representation. Therefore, the users are either required to explicitly provide their queries in a structured form, or to rely on a search engine to automatically convert their keyword queries into structured Indri queries.

The query hypergraph representation can be viewed as an instance of the latter option. Given an arbitrary keyword query, the query concepts and the dependencies between them are automatically identified and weighted using the query hypergraph induction process described in Chapter 3 of this dissertation. Then, the hypergraph representation is translated into the Indri query language using the language constructs described next. These queries are executed by the Indri search engine, and the results are presented to the user.

2.4.1 Concept Matching

Concepts are the basic building blocks of Indri queries. Concepts can come in the form of single terms, ordered or unordered phrases, synonyms, and wildcard expressions, among others. In addition, Indri allows the user to specify if a concept should appear within a certain field, or if it should be scored within a given context. In this section, we describe the subset of Indri concept matching operators used in this dissertation.

- t_1 — matches stemmed and normalized term t_1 .
- $\#N(t_1 t_2 \dots)$ — ordered window operator. Concept matches the document if terms in the window appear ordered, with at most $N - 1$ terms between each term.
- $\#\text{uwN}(t_1 t_2 \dots)$ — unordered window operator. Concept matches the document if terms in the window must appear within window of length N in any order.

While Indri allows the user to define a variety of concept scoring functions, in this dissertation we use its default scoring function, query likelihood with Dirichlet smoothing (ZHAI and LAFFERTY 2004)

$$f(\kappa, D) = \log \frac{tf(\kappa, D) + \mu \frac{tf(\kappa, \mathcal{C})}{|\mathcal{C}|}}{|D| + \mu}, \quad (2.1)$$

where κ is a concept defined by one of the operators described above; $tf(\kappa, D)$ and $tf(\kappa, \mathcal{C})$ are the number of concept occurrences in the document D and the collection, respectively; μ is a free parameter (set by default to 2,500); and $|D|$ is the length of the document D .

2.4.2 Belief Operators

Belief operators in the Indri query language allow the user to combine beliefs about concepts (i.e., concept scores). Indri provides both unweighted and weighted belief operators. With weighted operators, one can assign varying weights to certain expressions, to control the impact of each query concept on the final score. While Indri supports a variety of belief operators, in this section we describe only the subset pertinent to this dissertation.

- $\#\text{combine}(\kappa_1 \dots \kappa_n)$ — arithmetic mean of the concept scores of $\kappa_1, \dots, \kappa_n$

$$\#\text{combine}(\kappa_1 \dots \kappa_n) = \frac{\sum_{\kappa \in [\kappa_1, \dots, \kappa_n]} f(\kappa, D)}{n}$$

- $\#weight(w_1\kappa_1 \dots w_n\kappa_n)$ — weighted arithmetic mean of the concept scores $\kappa_1, \dots, \kappa_n$

$$\#weight(w_1\kappa_1 \dots w_n\kappa_n) = \sum_{\kappa_i \in [\kappa_1, \dots, \kappa_n]} \frac{w_i}{\sum_i w_i} f(\kappa_i, D)$$

- $\#max(\kappa_1 \dots \kappa_n)$ — maximum the concept scores of $\kappa_1, \dots, \kappa_n$

$$\#max(\kappa_1 \dots \kappa_n) = \max(f(\kappa_1, D) \dots f(\kappa_n, D)).$$

For each of the above belief operators, concepts $\kappa_1, \dots, \kappa_n$ are defined using one of the concept matching operators, and $f(\kappa, D)$ is defined in Equation 2.1.

2.4.3 Extents

Indri queries can be used to score and retrieve not only full documents, but document parts, or *extents*, as well. These extents can be fields in the document (for instance document title or anchor text), or document passages. To specify an extent retrieval, a field name in the square brackets is added to the belief operator.

For instance, for instantiating the $\#weight$ operator over fixed length passages of length L , with overlap O , one simply has to define an Indri query

$$\#weight[passageL:O](w_1\kappa_1 \dots w_n\kappa_n).$$

2.4.4 Combining Beliefs

The Indri query language is defined recursively. This enables further combination of the outcomes of the different belief operators.

For instance, to assign a score to a document based on the score of the entire document combined with the score of its highest scoring passage, one can use the following Indri query

```

#combine (
    #max(#weight [passageL:0] ( $w_1\kappa_1 \dots w_n\kappa_n$ ))
    #weight( $w_1\kappa_1 \dots w_n\kappa_n$ )
) .

```

2.5 Summary

In this chapter, we summarized the background and the previous work related to this dissertation. We described the unsupervised bag-of-words retrieval models (Section 2.1), as well as retrieval models that incorporate term dependencies (Section 2.2) and supervised term weighting (Section 2.3). Finally, in Section 2.4 we described the Indri structured query language, which is used for the experiments in this dissertation.

In the next chapter, we introduce query hypergraphs, a formal query representation framework that underlies all the retrieval models described in this dissertation. We fully describe the process of query hypergraph induction, query hypergraph parameterization and parameter optimization, and ranking with query hypergraphs.

CHAPTER 3

QUERY HYPERGRAPHS

In this chapter, we describe the proposed general theoretical framework for query representation and information retrieval based on *query hypergraphs*. We start the chapter with Section 3.1, in which we formally describe the process of representing search queries using a hypergraph structure. Then, in Section 3.2, we derive a ranking principle based on the query hypergraph representation. In Section 3.3 we describe the process of query hypergraph structure induction, and in Section 3.4 we describe the query hypergraph parameterization. In Section 3.5 we show how the parameters of the query hypergraphs are optimized. Finally, we conclude this chapter with Section 3.6.

3.1 Query Representation with Hypergraphs

In this chapter, we base the query representation on two primary modeling principles, which were first illustrated in Figure 1.2 in Chapter 1.

Principle 1 Given a query Q we can model it using a set of linguistic structures

$$\Sigma^Q = \{\sigma_1, \dots, \sigma_n\}.$$

The structures in the set Σ^Q are both *complete* and *disjoint*. The *completeness* of the structure implies that it can be used as an autonomous query representation. The *disjointness* of the structures means that there is no overlap in the linguistic phenomena modeled by the different structures. In other words, each structure groups together concepts of a single type (e.g., terms, bigrams, noun phrases, etc.).

Structure σ	Concepts $\{\kappa \kappa \in \sigma\}$
<i>Terms</i>	[“members”, “rock”, “group”, “nirvana”]
<i>Bigrams</i>	[“members rock”, “rock group”, “group nirvana”]
<i>Segments</i>	[“members”, “rock group”, “nirvana”]
<i>Named Entities</i>	[“nirvana”]
<i>Dependence</i>	[“members nirvana”, “rock group”]
<i>Corpus Expansion</i>	[“music”, “alternative”, “punk”, “bootlegs”]
<i>Wikipedia Expansion</i>	[“grohl”, “foo fighters”, “album”, “cobain”]

Table 3.1. Examples of the possible structures and the concepts they might contain for a search query “members rock group nirvana”.

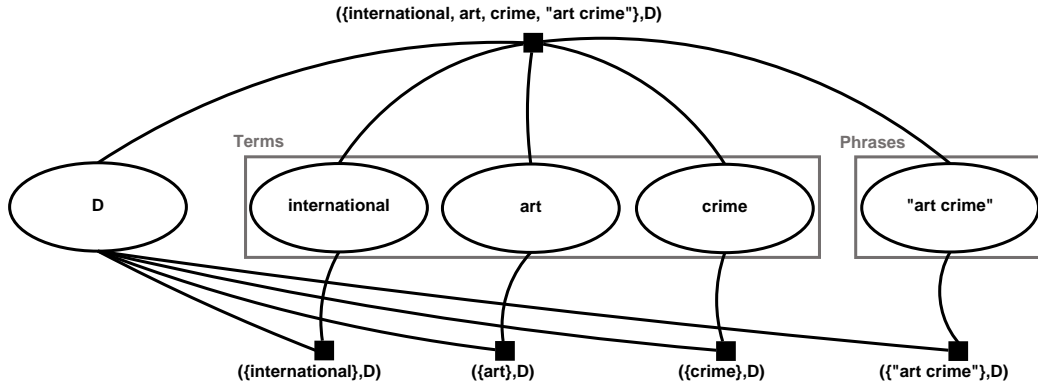


Figure 3.1. Example of a hypergraph representation for the query “international art crime”.

Principle 2 Within each structure arbitrary term dependencies can be modeled as concepts. In other words, each structure $\sigma_i \in \Sigma^Q$ is represented by a set of concepts

$$\sigma_i \triangleq \{\kappa_i^1, \kappa_i^2, \dots\}.$$

Each such concept is considered to be an atomic unit for the purpose of query representation. Note that in contrast to the view usually taken in information science (STOCK 2010) we do not require the concepts to carry a semantic meaning. Instead, we take an information retrieval centric approach and define a concept as an arbitrary syntactic expression that can be matched within a retrieved document.

In addition, for convenience, we adopt the notation

$$\kappa^Q \triangleq \bigcup_{i=1}^n \sigma_i,$$

to refer to the union of all the query concepts, regardless of their respective structures.

While the hierarchical query representation defined by the two principles described above is conceptually simple, it is flexible enough to allow a wide range of specific instantiations, and can model a large variety of linguistic structures that are often encountered in natural language processing and information retrieval applications. Table 3.1 illustrates how this query representation models a variety of concept types that can be extracted given the query “*members rock group nirvana*”, including terms, bigrams, query segments, named entities, dependencies and expansion terms.

Note that the hierarchical query representation is not limited to the explicit query terms alone. For instance, query expansion, a well known information retrieval technique, can also be modeled as a linguistic structure. The goal of query expansion is to enrich the original user query with additional related terms or concepts. This technique has been shown to be highly successful in various information retrieval applications (see, for instance, Chapter 6 in CROFT *et al.* (2009)). In the proposed query representation, query expansion is achieved by simply embedding the expanded concepts as a structure.

As an example, consider the query “*members rock group nirvana*” in Table 3.1. In addition to the structures based on the explicit query concepts that appear in Table 3.1, two structures based on expansion concepts are included as well. The first of these structures is based on the concepts that are expanded from the retrieval corpus, while the other structure is based on the concepts extracted from the Wikipedia. Expansion concepts from an additional source such as Wikipedia may prove beneficial, since they provide a complementary perspective on the query intent (e.g., compare the expansion concepts derived from the retrieval corpus to the concepts derived from the Wikipedia in Table 3.1).

Given the hierarchical query representation defined by the set of structures Σ^Q , our primary interest is in using it for modeling the relationship between a query Q and some document D in the corpus. Specifically, given a query representation Σ^Q and a document D , our aim is to construct a hypergraph $H(\Sigma^Q, D)$ ¹.

Most generally, a hypergraph is a generalization of a graph. A hypergraph H is represented by a tuple $\langle V, E \rangle$ such that

- V is a set of elements or *vertices*,
- E is a set of non-empty subsets of V , called *hyperedges*.

In other words, the set $E \subseteq \mathcal{PS}(V)$ of hyperedges is a subset of the powerset of V (KAUFMANN *et al.* 2009).

Specifically for the scenario of document retrieval, we define the hypergraph H over the document D and the set of query concepts κ^Q as

$$\begin{aligned} V &\triangleq \kappa^Q \cup \{D\} \\ E &\triangleq \{(\mathbf{k}, D) : \mathbf{k} \in \mathcal{PS}(\kappa^Q)\}. \end{aligned} \tag{3.1}$$

Figure 3.1 demonstrates an example of a hypergraph H for the search query “international art crime”. In this particular example, we have two structures. The first structure contains the query terms denoted i , a , and c , respectively. The second structure contains a single phrase, which we denote ac . Over these three concepts, we define a set of four hyperedges – one hyperedge connecting document D and *each of* the concepts, and one hyperedge connecting D and *all of* the concepts.

¹For conciseness, we use the abbreviation $H \triangleq H(\Sigma^Q, D)$ in the remainder of this dissertation.

Formally, for the hypergraph H in Figure 3.1, the vertices and the hyperedges are defined as follows

$$\begin{aligned} V_{\text{Fig. 2}} &= \{D, i, a, c, ac\} \\ E_{\text{Fig. 2}} &= \{(\{i\}, D), (\{a\}, D), (\{c\}, D), (\{i, a, c, ac\}, D)\}. \end{aligned}$$

Note that this hypergraph configuration is just one possible choice. In fact, any subset of query terms can serve as a query concept, and similarly, any subset of query concepts can serve as a hyperedge, as shown by Equation 3.1.

3.2 Ranking with Query Hypergraphs

In the previous section, we defined the query representation using a hypergraph $H = \langle V, E \rangle$. In this section, we define a global function over this hypergraph, which assigns a *relevance score* to document D in response to query Q . This relevance score is used to rank the documents in the retrieval corpus.

A factor graph, a form of hypergraph representation which is often used in statistical machine learning (BISHOP 2006), associates a factor ϕ_e with a hyperedge $e \in E$. Therefore, most generally, a relevance score of document D in response to query Q represented by a hypergraph H is given by

$$sc(Q, D) \triangleq \prod_{e \in E} \phi_e(\mathbf{k}_e, D) \stackrel{\text{rank}}{=} \sum_{e \in E} \log(\phi_e(\mathbf{k}_e, D)). \quad (3.2)$$

It is interesting to note that Equation 3.2 is reminiscent of the recently proposed log-linear retrieval models, including the Markov random field model (METZLER and CROFT 2005) and the linear discriminant model (GAO *et al.* 2005). Similarly to these models, Equation 3.2 scores a document using a log-linear combination of factors $\phi_e(\mathbf{k}_e, D)$.

However, an important difference from these retrieval models is related to the fact that the factors $\phi_e(\mathbf{k}_e, D)$ in Equation 3.2 are defined over *concept sets*, rather than *single concepts*, as in previous work (GAO *et al.* 2005; METZLER and CROFT 2005). This definition enables the modeling of higher-order dependencies between query terms. Higher-order term dependencies cannot be easily modeled by the existing retrieval models that incorporate term dependencies (GAO *et al.* 2005; LV and ZHAI 2009; METZLER and CROFT 2005; PARK *et al.* 2011; TAO and ZHAI 2007).

Thus far, we have provided only the most abstract definition of the query representation and ranking with query hypergraphs. In the remainder of this chapter, we provide an in-depth discussion of the query hypergraph induction and a more detailed derivation of the ranking function and its parameters.

First, in Section 3.3, we fully specify the structures, concepts, and hyperedges in the query hypergraph H . Then, in Section 3.4, we examine the different parameterizations of the ranking function based the query hypergraph H . Finally, in Section 3.5 we describe the procedures for ranking function parameter optimization.

3.3 Query Hypergraph Induction

3.3.1 Hypergraph Structures

There are many potential ways in which we could define the set of structures Σ^Q in the query hypergraph. In this dissertation, we focus on three types of structures that are successfully used in previous work on modeling term dependencies for information retrieval (BENDERSKY *et al.* 2010; BENDERSKY *et al.* 2011; METZLER and CROFT 2005; PENG *et al.* 2007). We leave a further exploration of other possible hypergraph structures to future work.

(1) *QT-structure*. The query term (QT) structure contains the individual query words t_i as concepts. Terms are the most commonly used concepts in information retrieval, both in bag-of-words models (PONTE and CROFT 1998; ROBERTSON and WALKER

1994) and models that incorporate term dependencies (METZLER and CROFT 2005; MISHNE and DE RIJKE 2005; GAO *et al.* 2005).

(2) *PH-structure*. The phrase (PH) structure contains the combinations of query terms that are matched as *exact phrases* in the document. Exact phrase matching has often been used for improving the performance of retrieval methods (FAGAN 1987; XU and CROFT 1996). Most recently, it has been shown that using query bigrams for exact phrase matching is a simple and efficient method for improving the retrieval performance in large scale web collections (BENDERSKY *et al.* 2010; BENDERSKY *et al.* 2011; METZLER and CROFT 2005; MISHNE and DE RIJKE 2005; PENG *et al.* 2007). Following this finding, we define the concepts in the PH-structure as adjacent query word pairs $(t_i t_{i+1})$.

(3) *PR-structure*. The PR-structure differs from the PH-structure in the way the concepts in the structure are matched in the document. In order to match the document, the individual terms in a concept in the PR-structure may occur in *any order* within a *window of fixed length*. In this dissertation, we fix the window size to $4|t|$ terms, where $|t|$ is the number of terms in the concept. This approach follows the definition of term proximity as defined by METZLER and CROFT (2005).

3.3.2 Hyperedges

As described in Section 3.1, a naïve induction approach may result in an exponential number of hyperedges in a query hypergraph. This is due to the fact that each hyperedge e can model a dependency between an arbitrary subset of concepts. Thus, theoretically, we could define $E \triangleq \mathcal{PS}(\kappa^Q)$. Such an approach would be detrimental for two reasons.

First, for efficiency reasons, the naïve approach would result in a significantly increased query latency, especially for verbose natural language queries which are the

focus of this dissertation. This is due to the fact that the cardinality of the set of the hyperedges E would grow exponentially with the size of the query.

Second, modeling dependencies between each subset of the query concepts could be detrimental for the retrieval effectiveness as well. Most of these dependencies are redundant, and some might actually hurt the retrieval effectiveness by introducing intents that are not aligned with the true query intent. For instance, consider a dependency between the concepts “*crime*” and “*international crime*” for the query “*international art crime*” in Figure 3.1. Such a dependency could be beneficial for a broad query about international crime, but not for a query focused on art crime.

Therefore, in this dissertation we limit our attention to only two types of hyperedges. Both of these types of hyperedges have an intuitive appeal from the information retrieval perspective.

(1) Local hyperedges. For each concept $\kappa \in \kappa^Q$, we define a hyperedge $(\{\kappa\}, D)$. This local edge² represents the contribution of the concept κ to the total document relevance score, regardless of the other query concepts. As we show in Section 3.3.3.1, the factors defined over the local edges are akin to the functions that are usually employed in the existing log-linear retrieval models (GAO *et al.* 2005; METZLER and CROFT 2005).

(2) Global hyperedge. In addition to the local edges, we define a *single* global hyperedge (κ^Q, D) over the entire set of query concepts κ^Q . This global hyperedge provides the evidence about the contribution of each concept $\kappa \in \kappa^Q$ given its dependency on the entire set of query concepts κ^Q . Unlike in the case of local edges, the factors defined over the global hyperedge cannot be easily expressed using the existing

²From now on, we refer to the local hyperedges simply as *edges*, since they are defined over a vertex pair, rather than an arbitrary set of vertices.

log-linear retrieval models, and draw inspiration from prior work on passage-based retrieval. These factors are described in Section 3.3.3.2.

Figure 3.1 provides a simple example of these two types of hyperedges. The hyperedges at the bottom of the hypergraph in Figure 3.1 are the local edges, while the hyperedge at the top is the global hyperedge.

3.3.3 Factors $\phi_e(\mathbf{k}_e, D)$

Following the hyperedge induction process described in Section 3.3.2, in this section we define two types of factors. The local factors – corresponding to the local edges – are defined in Section 3.3.3.1; the global factor – corresponding to the global hyperedge – is defined in Section 3.3.3.2.

Both local and global factors incorporate a *matching function* $f(\kappa, X)$, which assigns a score to the occurrences of the concept κ in a text fragment X . This function may take various forms, however in information retrieval applications it is commonly a monotonic function, i.e., its value increases with the number of times concept κ matches document D .

As a matching function, following some previous work on log-linear retrieval models (BENDERSKY *et al.* 2010; GAO *et al.* 2005; METZLER and CROFT 2005), we use a log of the language modeling estimate for concept κ with Dirichlet smoothing (ZHAI and LAFFERTY 2004), i.e.

$$f(\kappa, X) \triangleq \log \frac{tf(\kappa, X) + \mu \frac{tf(\kappa, \mathcal{C})}{|\mathcal{C}|}}{\mu + |X|}, \quad (3.3)$$

where $tf(\kappa, X)$ and $tf(\kappa, \mathcal{C})$ are the number of occurrences of the concept κ in the text fragment and the collection, respectively; μ is a free parameter; $|X|$ is the number of terms in X , and $|\mathcal{C}|$ is the total number of terms in the collection.

We use this language modeling estimate as a concept matching function since it is convenient and efficient to compute, and exhibits state-of-the-art retrieval performance in other concept-based retrieval models (BENDERSKY *et al.* 2010; GAO *et al.* 2005; METZLER and CROFT 2005). However, other commonly used matching functions (such as BM25 (ROBERTSON and WALKER 1994) or DFR (AMATI and VAN RIJSBERGEN 2002)) can be substituted in Equation 3.3 without loss of generality.

3.3.3.1 Local Factors

The local factors are defined over the local edges $(\{\kappa\}, D)$. A local factor assigns a score to the occurrences of concept κ in the document D , regardless of the other query concepts. Therefore, a local factor is defined similarly to the previously proposed log-linear retrieval models (BENDERSKY *et al.* 2010; GAO *et al.* 2005; METZLER and CROFT 2005)

$$\phi(\{\kappa\}, D) \triangleq \exp\left(\lambda(\kappa)f(\kappa, D)\right), \quad (3.4)$$

where $\lambda(\kappa)$ is an importance weight assigned to the concept κ , and $f(\kappa, D)$ is a matching function between the concept κ and the document D .

Using the Indri query language (described in Section 2.4), all the local factors can be combined into a structured Indri query of the following form

$$\#weight(w_1\kappa_1 \dots w_n\kappa_n).$$

3.3.3.2 The Global Factor

The global hyperedge (κ^Q, D) described in Section 3.3.2, represents a dependency between the entire set of query concepts. In this section, we present a global factor that is defined over this hyperedge.

*...Simi Valley, West Covina and Los Angeles police departments were among the first **law enforcement** agencies to receive money through the forfeiture program....a narcotics-sniffing **dog** in a Simi Valley police investigation...led to the largest seizure of cocaine ever by authorities from Ventura County...**dog**'s efforts are expected to yield a substantial amount of money...for the 21-officer department...*

Figure 3.2. Excerpt a relevant document retrieved in response to the query “Provide information on the use of dogs worldwide for law enforcement purposes”. Non-stopword query terms are marked in boldface.

A common way to estimate a dependency between query terms is using a measure of their proximity in a retrieved document (CUMMINS and O’RIORDAN 2009; LV and ZHAI 2009; METZLER and CROFT 2005; TAO and ZHAI 2007). Analogously, we may simply choose to estimate a dependency between query concepts using similar proximity measures. However, there are two notable difficulties that impede an application of this approach to concept dependency.

First, the existing term proximity measures usually capture close, sentence-level, co-occurrences of the query terms in a retrieved document (METZLER and CROFT 2005; PENG *et al.* 2007; TAO and ZHAI 2007). The dependency range is much longer for concept dependencies. For instance, in the example in Figure 3.2, the concepts *dog* and *law enforcement* do not ever appear in the same sentence. However, the dependency between them is revealed when examining their co-occurrences in a larger text passage.

Second, since concepts can be arbitrarily complex syntactic expressions, the probability of observing a *concept co-occurrence* is much lower than the probability of observing a *term co-occurrence*, even in large collections. For instance, most documents in the retrieved list for the query in Figure 3.2, do not contain both of the concepts *dog* and *law enforcement* in a context of a single passage.

Therefore, instead of estimating the dependency between query concepts using the standard proximity measures, we leverage a long history of research on passage retrieval (BENDERSKY and KURLAND 2008; CAI *et al.* 2004; CALLAN 1994; LIU and CROFT 2004; KASZKIEL and ZOBEL 1997; WANG and SI 2008; WILKINSON 1994) for the derivation of the global factor.

In the passage retrieval literature, a document is often segmented into overlapping passages of text of fixed size (KASZKIEL and ZOBEL 1997; KASZKIEL and ZOBEL 2001). The document is then scored using some combination of document-level and passage-level scores. One of the most successful and frequently-used score combinations is the **Max-Psg** combination, which uses the highest scoring passage to assign a score to the document (BENDERSKY and KURLAND 2008; CAI *et al.* 2004; KASZKIEL and ZOBEL 1997; LIU and CROFT 2002; WILKINSON 1994).

Similarly to the **Max-Psg** retrieval model, we define the global factor using a passage π , which receives the highest score among the set Π_D of passages extracted from the document D . Formally,

$$\phi(\boldsymbol{\kappa}^Q, D) \triangleq \exp \left(\max_{\pi \in \Pi_D} \sum_{\kappa \in \boldsymbol{\kappa}^Q} \lambda(\kappa, \boldsymbol{\kappa}^Q) f(\kappa, \pi) \right), \quad (3.5)$$

where $\lambda(\kappa, \boldsymbol{\kappa}^Q)$ is the importance weight of the concept κ in the context of the entire set of query concepts $\boldsymbol{\kappa}^Q$, and $f(\kappa, \pi)$ is a matching function between the concept κ and a passage $\pi \in \Pi_D$.

Intuitively, the global factor in Equation 3.5 assigns a higher relevance score to a document that contains many important concepts in the confines of a single passage. Note that the importance weight $\lambda(\kappa, \boldsymbol{\kappa}^Q)$ of a concept in the global factor is determined not only by the concept itself – as in the case of the importance weights $\lambda(\kappa, D)$ in the local factors – but also by the concepts that co-occur together with the concept in the passage π .

Using the Indri query language (described in Section 2.4), the global factor can be formulated using a structured Indri query of the following form

$$\#max(\#weight[passageL:0](w_1\kappa_1 \dots w_n\kappa_n)).$$

3.4 Query Hypergraph Parameterization

In the previous section, we introduced two types of concept weights that parameterize the ranking function in Equation 3.2. First, there are the independent importance weights $\lambda(\kappa)$ that parameterize the local factors (see Equation 3.4). Second, there are the importance weights $\lambda(\kappa, \boldsymbol{\kappa}^Q)$ that assign weight to a concept, while taking into account the rest of the concepts in the query (see Equation 3.5).

In this section, we consider two possible parameterization schemes for these concept weights. In Section 3.4.1, we consider parameterization by structure. Conversely, in Section 3.4.2, we examine parameterization by concept.

3.4.1 Parameterization By Structure

A simple way to parameterize the importance weights $\lambda(\kappa)$ and $\lambda(\kappa, \boldsymbol{\kappa}^Q)$, is to make the assumption that the weights of all the concepts in the same structure are tied. Formally:

$$\begin{aligned} \forall \kappa_i, \kappa_j \in \boldsymbol{\sigma} & : \lambda(\kappa_i) = \lambda(\kappa_j) = \lambda(\boldsymbol{\sigma}) \\ \forall \kappa_i, \kappa_j \in \boldsymbol{\sigma} & : \lambda(\kappa_i, \boldsymbol{\kappa}^Q) = \lambda(\kappa_j, \boldsymbol{\kappa}^Q) = \lambda(\boldsymbol{\sigma}, \Sigma^Q) \end{aligned}$$

This assumption has the benefit of significantly reducing the number of free parameters in the retrieval model, thereby greatly simplifying the estimation process. Due to its simplicity, parameterization by structure is commonly used in the log-linear retrieval models (GAO *et al.* 2005; METZLER and CROFT 2005; PENG *et al.* 2007).

Feature	Type	Description
CF(κ)	Endogenous	Frequency of κ in the collection
DF(κ)	Endogenous	Document frequency of κ in the collection
GF(κ)	Exogenous	Frequency of κ in Google n-grams
WF(κ)	Exogenous	Frequency of κ in Wikipedia titles
QF(κ)	Exogenous	Frequency of κ in a search log
AP(κ)	Constant	A priori constant weight (=1)

Table 3.2. Concept importance features Φ .

Using parameterization by structure and the definitions of local and global factors in Section 3.3.3, we can explicitly rewrite the ranking function in Equation 3.2 as

$$\begin{aligned}
sc(Q, D) &= \sum_{\sigma \in \Sigma^Q} \lambda(\sigma) \sum_{\kappa \in \sigma} f(\kappa, D) + \\
&+ \max_{\pi \in \Pi_D} \sum_{\sigma \in \Sigma^Q} \lambda(\sigma, \Sigma^Q) \sum_{\kappa \in \sigma} f(\kappa, \pi). \tag{3.6}
\end{aligned}$$

3.4.2 Parameterization By Concept

The main drawback of parameterization by structure is the fact that it implies that all the concepts in the same structure are equally important for expressing the query intent. This implication is not always true, especially for more verbose, grammatically complex queries, which may benefit from assigning varying concept weights (BENDERSKY and CROFT 2008; BENDERSKY *et al.* 2010; LEASE *et al.* 2009).

Therefore, we may wish to remove the restriction imposed in the previous section, and parameterize the concept weights based on the concepts themselves rather than their respective structures. Assigning a single weight to each concept is clearly infeasible, since the number of concepts is exponential in the size of the vocabulary. Therefore, we take a parameterization approach and represent each concept using a combination of *importance features*, Φ , described in Table 3.2. These importance features are based on concept frequencies, and can be efficiently computed and cached, even for large-scale collections.

The features in the Table 3.2 are computed for each concept κ (as defined in Section 3.3.1) and are independent of a specific document. This fact allows us to combine the statistics of the underlying document corpus with the statistics of various external data sources to achieve a potentially more accurate weighting. Accordingly, we divide the features used for concept importance weighting into two main types, based on the type of information they are using.

The first type, the *endogenous*, or collection-dependent, features are akin to standard weights used in information retrieval. They are based on collection frequency counts and document frequency counts calculated over a particular document corpus on which the retrieval is performed.

The second type, the *exogenous*, or collection-independent, features are calculated over an array of external data sources. The use of such sources was found to be beneficial for information retrieval models in previous work (BAI *et al.* 2008; BENDERSKY and CROFT 2008; LEASE *et al.* 2009). Some of these data sources provide better coverage of terms, and can be used for smoothing sparse concept frequencies calculated over smaller document collections. Others provide more focused sources of information for determining concept importance. In this dissertation, we use three external data sources: (i) a large collection of web n -grams, (ii) a sample of a query log, and (iii) Wikipedia. Although there are numerous additional data sources that could be potentially used, we intentionally limit our attention to these three sources as they are available for research purposes, and can be easily used to reproduce the reported results.

The first source, *Google n-grams* corpus, is available from the Linguistic Data Consortium catalog (BRANTS and FRANZ 2006). The Google n -grams corpus contains the frequency counts of English n -grams generated from approximately 1 trillion word tokens of text from publicly accessible Web pages. We expect these counts to provide a

more accurate frequency estimator, especially for smaller corpora, where some concept frequencies may be underestimated due to the collection size.

In addition, we use a large sample of a query log consisting of approximately 15 million queries, which is available as a part of Microsoft 2006 RFP dataset³. We use this data source to estimate how often a concept occurs in user queries. Intuitively, we assume a positive correlation between an importance of a concept for retrieval and the frequency with which it occurs in queries formulated by the search engine users.

Finally, our third external data source is a snapshot of Wikipedia article titles⁴. Due to the large volume and the high diversity of topics covered by Wikipedia (as of April 2011, there are close to 8.5 million articles in English alone), we assume that the important concepts will often appear as (a part of) article titles in Wikipedia.

Table 3.2 details the statistics used for computing the concept importance features. The statistics presented in the Table 3.2 are computed for each of the concepts defined by the query structures (QT,PH and PR – see Section 3.3.1 for details). Using the set of importance features Φ based on these statistics, we can parameterize the importance weights $\lambda(\kappa)$ and $\lambda(\kappa, \kappa^Q)$ as

$$\begin{aligned} \forall \kappa \in \sigma \quad : \quad \lambda(\kappa) &= \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \varphi(\kappa) \\ \forall \kappa \in \sigma \quad : \quad \lambda(\kappa, \kappa^Q) &= \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma, \Sigma^Q) \varphi(\kappa). \end{aligned}$$

Note that this concept weight parameterization requires us to compute parameters based on importance features and structures, rather than the concepts themselves. This approach makes the parameterization by concept approach feasible, since we are no longer required to compute an individual parameter for each concept in the

³See <http://research.microsoft.com/en-us/um/people/nickcr/wscd09/> for more details about this dataset.

⁴Available at: <http://download.wikimedia.org/enwiki/>

vocabulary. Instead, the cardinality of the free parameters vector Λ (which includes both the local factor parameters $\lambda(\kappa)$ and the global factor parameters $\lambda(\kappa, \kappa^Q)$) is reduced down to

$$|\Lambda| = 2|\Sigma^Q||\Phi| = 2 \cdot 3 \cdot 6 = 36.$$

Using the importance features for concept weight parameterization, we can explicitly rewrite the ranking function in Equation 3.2 as

$$\begin{aligned} sc(Q, D) &= \sum_{\sigma \in \Sigma^Q} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) + \\ &+ \max_{\pi \in \Pi_D} \sum_{\sigma \in \Sigma^Q} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma, \Sigma^Q) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, \pi). \end{aligned} \tag{3.7}$$

3.5 Parameter Optimization

In this section, we describe the optimization of the parameters used in the query hypergraph ranking function. First, in Section 3.5.1, we discuss the general learning-to-rank paradigm for information retrieval, and how it differs from the query hypergraph parameterization. Then, in Section 3.5.2, we describe coordinate ascent – a simple yet effective parameter optimization technique used as a base procedure in the query hypergraph parameter optimization. Finally, in Section 3.5.3, we describe the pipeline approach to query hypergraph parameter optimization.

3.5.1 Learning To Rank

Learning to rank (LTR) has recently become a popular paradigm for optimizing the ranking of documents in information retrieval, especially in the setting of web search (LI 2011; BURGESS *et al.* 2005; JOACHIMS 2002). Most generally, the goal of the standard LTR techniques is to learn an optimally relevant ranking of the document set \mathbf{D} in response to a set of training queries \mathbf{Q} .

Formally, for each query $Q_i \in \mathbf{Q}$, a list of documents $\{D_i^1, \dots, D_i^n\} \in \mathbf{D}$ is derived (e.g., using a standard retrieval technique such as BM25). Then, each query-document pair $\langle Q_i, D_i^j \rangle$ is associated with a relevance label L_i^j and with a feature vector Ψ_i^j . This feature vector commonly includes features based on the query-document text match scores, link-based features and query-based features (LI 2011).

Once each query-document pair $\langle Q_i, D_i^j \rangle$ is associated with a feature vector Ψ_i^j and a relevance label L_i^j , a variety of machine learning techniques including support vector machines (JOACHIMS 2002), ordinal regression (LI *et al.* 2007), neural networks (BURGES *et al.* 2005), boosting (XU and LI 2007) and bagging (MOHAN *et al.* 2011) can be employed to learn the scoring function $sc(Q_i, D_i^j)$ that is trained to optimize some rank-based criteria (for instance, normalized discounted cumulative gain or average precision) such that the documents with higher relevance labels appear higher in the ranked list.

The LTR setting bears a close similarity to the problem of parameter optimization in query hypergraphs. In both the LTR and the query hypergraph settings, the parameters of the scoring function are learned such that the relevance of the resulting ranking is optimized.

However, the main difference between the LTR and the query hypergraph settings lies in the choice of the parameterization of the ranking function. In the LTR setting, the scoring function is parameterized based on the features defined over a query-document pair. In contrast, in the setting of the query hypergraph, the scoring function is parameterized based on the features defined over arbitrary subsets of query concepts (see Section 3.2).

It is interesting to note that the LTR and the query hypergraph optimization approaches are complementary. While the former is focused on optimizing the ranking of a given set of documents \mathbf{D} , the latter is focused on deriving this document set

```

CoordinateAscent( $\mathcal{I}, \Lambda^0$ )
1:  $\Lambda \leftarrow \Lambda^0$ 
2:  $\mathcal{M} \leftarrow \text{eval}(\mathcal{I}, \Lambda)$ 
3: change  $\leftarrow$  TRUE
4:  $i \leftarrow 0$ 
5: while change and  $i \leq \text{MAX\_ITER}$  do
6:   for  $\lambda \in \Lambda$  do
7:      $\lambda' \leftarrow \text{optimize}(\lambda, \mathcal{I}, \Lambda)$ 
8:     if  $\lambda' \neq \lambda$  then
9:       update( $\Lambda, \lambda'$ )
10:       $\mathcal{M} \leftarrow \text{eval}(\mathcal{I}, \Lambda)$ 
11:      change  $\leftarrow$  TRUE
12:     else
13:       change  $\leftarrow$  FALSE
14:     end if
15:   end for
16:    $i \leftarrow i + 1$ 
17: end while
18: return  $\langle \mathcal{M}, \Lambda \rangle$ 

```

Figure 3.3. The outline of the coordinate ascent optimization algorithm.

(e.g., for replacing a standard retrieval technique such as BM25 for constructing this document set).

LTR approaches are generally classified into *pointwise*, *pairwise* and *listwise* (LI 2011). In this dissertation, we use a simple *listwise* approach which directly optimizes a metric of interest and is effective and efficient, especially for a small set of parameters as described in this work. This technique is called *coordinate ascent* and was first proposed by METZLER and CROFT (2007b). It is further described in Section 3.5.2.

Finally, in some cases the hypergraph optimization can be done in several stages, in a pipeline fashion where each optimization step feeds into the next stages. This process is further detailed in Section 3.5.3.

3.5.2 Coordinate Ascent

Note that the local and the global factors in Equation 3.4 and Equation 3.5, respectively, are linear with respect to the set of free parameters Λ , which is based either on structures (see Section 3.4.1) or concepts (see Section 3.4.2). Therefore, as a base algorithm for optimizing the scoring function parameters, we make use of the coordinate ascent (CA) algorithm proposed by METZLER and CROFT (2007b).

Figure 3.3 outlines the CA algorithm. As an input, the CA algorithm receives (a) a set of fixed parameters \mathcal{I} (which may be an empty set) that will not be updated by the algorithm, and (b) an initial parameter set Λ^0 , which may be initialized to random values or set based on some prior knowledge.

The CA algorithm iteratively optimizes a target metric \mathcal{M} (in our case a retrieval effectiveness metric such as average precision). This optimization is done by performing a one-dimensional optimization using a line search for each of the parameters $\lambda \in \Lambda$ (represented by the **optimize** function in Figure 3.3), while holding the other parameters fixed. This cycle of one-dimensional optimizations is repeated as long as both of the two conditions are met (the **while** loop in Figure 3.3):

- (a) At least one of the parameters λ is changed during the cycle (i.e., the metric \mathcal{M} improved during the cycle as determined by the **eval** function).
- (b) Number of iterations did not reach the maximum number of allowed iterations `MAX_ITER`.

While CA is a simple optimization algorithm, it has several advantages that justify its use for query hypergraph optimization. First, it directly optimizes the retrieval metric, therefore sidestepping the metric divergence problem, which is common in the other learning to rank methods (METZLER 2007a). Second, CA is efficient, especially for a small number of parameters, since the algorithm runtime is bounded by $|\Lambda|\text{MAX_ITER}$. Finally, coordinate ascent has been shown to perform well for a

```

PipelineOptimization( $\Lambda^0$ )
1:  $\mathcal{I} \leftarrow \emptyset$ 
2: for  $\Lambda_i^0 \in \Lambda^0$  do
3:    $\langle \mathcal{M}, \Lambda_i \rangle \leftarrow \text{CoordinateAscent}(\mathcal{I}, \Lambda_i^0)$ 
4:    $\mathcal{I} \leftarrow \mathcal{I} \cup \{\Lambda_i\}$ 
5: end for
6: return  $\langle \mathcal{M}, \mathcal{I} \rangle$ 

```

Figure 3.4. The outline of the pipeline optimization.

variety of LTR tasks in prior work (METZLER and CROFT 2007b; METZLER 2007a; METZLER 2007b; DANG and CROFT 2010). The empirical results presented in this dissertation further validate the effectiveness of the coordinate ascent method.

3.5.3 Pipeline Optimization

While the base optimization procedure using coordinate ascent (as described in Section 3.5.2) assumes that the optimized function is linear in the set of parameters Λ , in practice, some types of query hypergraphs would require multiple stages of optimization. For instance, for query expansion, we first need to optimize the parameters of the query hypergraph based on the explicit query concepts, and then use the optimized hypergraph to expand the query, and build a new query hypergraph including the expansion concepts.

In this section, we give a high-level overview of how we handle the multi-stage parameter optimization in query hypergraphs. To this end, we employ a simple *pipeline* algorithm. While other joint optimization techniques are available in machine learning and natural language processing literature (see FINKEL (2010) for a detailed overview), we choose the pipeline algorithm, since it is conceptually simple and efficient, and produces good empirical results.

Figure 3.4 describes the pipeline optimization algorithm. Given n free parameter sets, $\Lambda = \langle \Lambda_1, \dots, \Lambda_n \rangle$, the parameter sets are optimized sequentially using the coordinate ascent algorithm (see Figure 3.3). At the i -th stage of the optimization

sequence, the previously optimized parameter sets $\langle \Lambda_1, \dots, \Lambda_{i-1} \rangle$ are held fixed, while the parameter set Λ_i is being optimized. At the end of the optimization procedure, the entire set of parameters $\mathbf{\Lambda}$ is optimized.

Note that the pipeline optimization algorithm does not necessarily reach a global optimum, since the parameters are optimized sequentially, and no updates are applied to the parameters $\langle \Lambda_1, \dots, \Lambda_{i-1} \rangle$, when the parameter set Λ_i is added to the sequence. In practice, however, we found that pipeline optimization avoids overfitting, and achieves a good empirical performance. We hypothesize that this is due to the fact that only a small set of parameters is optimized at each step in the sequence, which improves the effectiveness of the coordinate ascent algorithm.

3.6 Summary

In this chapter, we presented the theoretical foundations of query representation using query hypergraphs. First, in Section 3.1, formally described the process of representing search queries using a hypergraph structure. Then, in Section 3.2, we derived a ranking principle based on the query hypergraph representation. In Section 3.3 we described the process of query hypergraph structure induction, and in Section 3.4 we described the query hypergraph parameterization. Finally, in Section 3.5 we fully specified the process of the pipeline optimization of the parameters in a query hypergraphs.

In the next chapters of this dissertation we will describe practical implementations of retrieval models based on the theoretical query hypergraph framework. First, in Chapter 4 we will describe the datasets and the retrieval metrics used for empirical evaluation of our retrieval models. Then, in Chapter 5, we focus on parameterized concept weighting in the query hypergraph framework. In Chapter 6 and Chapter 7 we focus on query expansion with query hypergraphs. Finally, in Chapter 8 we focus on modeling parameterized concept dependencies using query hypergraphs.

CHAPTER 4

DATASETS AND EVALUATION

In the chapters that follow, we describe the experimental evaluation of the different retrieval models based on query hypergraphs. Therefore, in this chapter, we detail the experimental setup used in the remainder of this dissertation. In Section 4.1 we describe the TREC corpora we use for the evaluation. Then, in Section 4.2 we outline the evaluation criteria used to measure the performance of the retrieval models.

4.1 TREC Corpora

The Text REtrieval Conference (TREC) series has produced a number of test corpora over the years. These test corpora are extensively used by the information retrieval community to enable the advancement of the state-of-the-art in retrieval models and to ensure the reproducibility of the experimental results published in major academic conferences. More details about TREC can be found at <http://trec.nist.gov/>.

Table 4.1 summarizes the three TREC corpora used in this dissertation. As can be seen from Table 4.1, these corpora vary by type, number of documents, and number of relevant judgments, thereby providing a diverse experimental setup for assessing the robustness of the proposed retrieval models.

Each of the TREC corpora in Table 4.1 consists of a document collection, a set of topics and a corresponding set of relevance judgments. In the next sections, we describe each of them in more detail.

Name	Description	# Docs	# Topics	# Rel. Judg.
<i>Robust04</i>	528,155	Newswire	250	311,409
<i>Gov2</i>	25,205,179	.gov domain crawl	150	135,352
<i>ClueWeb-B</i>	50,220,423	Web crawl	100	28,963

Table 4.1. Summary of TREC document collections, topics and relevance judgments used for evaluation.

$\langle id \rangle$	53
$\langle title \rangle$	discovery channel store
$\langle desc \rangle$	Find locations and information about Discovery Channel stores and types of products they sell.

Figure 4.1. An example of $\langle title \rangle$ and $\langle desc \rangle$ queries in a TREC topic §53.

4.1.1 Document Collections

As evident from Table 4.1, the definition of a *document* in a TREC corpus depends on the origin and the purpose of the corpus. Document definitions may range from news articles in traditional newswire corpora, to emails in enterprise search corpora¹, to, most recently, tweets in microblog corpora².

In the experiments in this dissertation, for the newswire corpus *Robust04*, the documents in the collection are news articles from different sources (e.g., Financial Times or LA Times). For the *Gov2* corpus, the documents in the collection are web pages collected in the crawl of the .gov domain conducted in 2004. Finally, for the largest corpus, *ClueWeb-B*, the documents are web pages with the highest crawl priority derived from a large general English-language web corpus.

4.1.2 Topics

In addition to the document collection, the TREC corpora contains a set of pre-defined topics, which can viewed as representations of information needs that users

¹<http://www.ins.cwi.nl/projects/trec-ent/>

²<http://trec.nist.gov/data/tweets/>

Grade	Label	Description
3	<i>Key</i>	This page or site is dedicated to the topic; authoritative and comprehensive, it is worthy of being a top result in a web search engine.
2	<i>HRel</i>	The content of this page provides substantial information on the topic.
1	<i>Rel</i>	The content of this page provides some information on the topic, which may be minimal; the relevant information must be on that page, not just promising-looking anchor text pointing to a possibly useful page.
0	<i>Non</i>	The content of this page does not provide useful information on the topic, but may provide useful information on other topics, including other interpretations of the same query.
-2	<i>Junk</i>	This page does not appear to be useful for any reasonable purpose; it may be spam or junk.

Table 4.2. Graded relevance scale for the *ClueWeb-B* corpus.

may have, given the collection of documents in the corpus. The contents of the information needs or topics depend on the nature of the underlying TREC corpus. For instance, for a web corpus *ClueWeb-B*, the topics are general informational inquiries, while for the more specialized *Gov2* corpus they focus on themes related to governance.

Each topic consists of a $\langle title \rangle$ and a $\langle desc \rangle$ query. The $\langle title \rangle$ and the $\langle desc \rangle$ queries in each topic represent the same information need, but differ in their level of verbosity. A $\langle title \rangle$ query is a short keyword query, while a $\langle desc \rangle$ query is a verbose natural language description of the information need. Figure 4.1 shows an example of $\langle title \rangle$ and $\langle desc \rangle$ queries for a standard TREC topic.

In the experiments in this dissertation, we treat the $\langle title \rangle$ and the $\langle desc \rangle$ queries as two separate query sets. In this way, we are able to demonstrate the performance of the proposed retrieval methods for both keyword and verbose queries and to assess their robustness across different types of document collections and query types.

4.1.3 Relevance Judgments

To assess how much relevant documents can be retrieved from the collection in response to each of the topics, a TREC corpus also provides a set of documents that are manually judged for relevance. Different definitions and scales of relevance are used for different tasks. For newswire collections such as *Robust04*, binary relevance judgments (*relevant* vs. *non-relevant*) are used. For web collections, documents are judged on a graded scale. For instance, for the web collection *ClueWeb-B*, the relevance scale has five grades that are shown in Table 4.2.

Note that it is easy to convert a graded relevance judgment into a binary relevance judgment. For instance, for the scale in Table 4.2, grades $\langle 3, 2, 1 \rangle$ will be mapped to the *relevant* label, and grades $\langle 0, -2 \rangle$ will be mapped to the *non-relevant* label.

The relevance judgments are used as the “ground truth” for the purposes of retrieval evaluation. In the next section, we describe how this evaluation is conducted.

4.2 Evaluation

4.2.1 Binary Evaluation Metrics

Recall that in the case of binary relevance judgments, for a given query Q the set of relevance judgments \mathcal{R} consists of labeled documents, where the label of i -th document is $R_i \in \{0 - \text{non-relevant}, 1 - \text{relevant}\}$.

Given this set, we can evaluate the retrieval performance using standard classification metrics, i.e. precision and recall. However, retrieval systems return a ranked list of documents, and users might only be interested in examining this list until a certain cutoff k is reached. For instance, in the case of web search, users are likely to stop examining the ranked list when reaching the end of the first page of search results (i.e., $k = 10$).

Therefore, one popular retrieval evaluation metric that we report in this dissertation is *precision at k -th result*, which is defined as

$$P@k = \frac{\sum_{i=1}^k R_i}{k}.$$

However, since $P@k$ only takes into account the top k retrieved results, and ignores the rest of the ranked list, we also use the *average precision* metric. Average precision can be thought of as a weighted precision measure that gives higher weight to relevant documents that appear near the top of the ranked list. The measure is computed by averaging $P@k$ for every position k where a relevant document is retrieved, up to a depth of 1,000 documents. The average precision measure implicitly accounts for both precision and recall, and is typically used to evaluate retrieval tasks where both precision and recall are important factors.

Formally, average precision is defined as

$$AP = \frac{\sum_{k: R_k=1} P@k}{|\mathcal{R}|}.$$

4.2.2 Graded Evaluation Metrics

For TREC web corpora that contain graded relevance judgments (as shown in Table 4.2), it is more suitable to compute metrics that take the grades of the relevance judgments into account rather than just their binary values.

One such metric that we report in this dissertation is *normalized discounted cumulative gain* at rank k ($NDCG@k$), which was first proposed by JÄRVELIN and KEKÄLÄINEN (2002). Using a set of graded relevance judgments \mathcal{R} for the query Q , $NDCG@k$ measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list down to the position k such that the gain of the results is discounted at lower ranks.

$NDCG@k$ is defined as

$$NDCG@k = \frac{1}{Z_{\mathcal{R}}} \sum_{i=1}^k \frac{2^{R_i} - 1}{\log_2(i + 1)},$$

where $Z_{\mathcal{R}}$ is a normalizing constant that is computed using an ideal ordering of the documents in the ranked list.

Another graded relevance metric that we report in this dissertation is *expected reciprocal rank* ($ERR@k$), which was recently proposed by CHAPELLE *et al.* (2009). $ERR@k$ is based on the cascade user browsing model (CRASWELL *et al.* 2008), which assumes that a user scans through ranked search results in order, and for each document, evaluates whether the document satisfies the query, and if it does, stops the search. Expected reciprocal rank is then defined as the expectation of the reciprocal rank of a result at which a user stops.

First, the probability of user being satisfied with the i -th result is defined as

$$P_i = \frac{2^{R_i} - 1}{2^{R_{\max}}},$$

where R_{\max} is the highest scale of the graded relevance judgments. Using this definition of probability P_i , the expected reciprocal rank is computed as

$$ERR@k = \sum_{i=1}^k \frac{P_i}{i} \prod_{j=1}^{i-1} (1 - P_j).$$

CHAPELLE *et al.* (2009) showed that $ERR@k$ consistently correlates better with a wide range of click-based metrics compared to $NDCG@k$ and other editorial metrics. The difference in correlation is particularly pronounced for navigational, short, and head queries.

4.2.3 Statistical Significance

The setup of most information retrieval experiments is as follows. We are given two retrieval systems: baseline system B , and some candidate system A . We need to determine whether the candidate retrieval system A is indeed better than a baseline retrieval system B , as hypothesized, and whether this difference is *statistically significant*. To determine statistically significant difference, it is not sufficient to compare

the average of some graded or binary retrieval metric (such as AP or $NDCG@K$) across all the queries. Instead, the candidate system A and the baseline system B are compared using one of the standard statistical significance methods.

There is an array of statistical significance testing methods that can be used to compare systems A and B , including Wilcoxon signed rank, sign test, Student's t-test and others. Please refer to SMUCKER *et al.* (2007) for a detailed evaluation of these statistical significance methods.

In this dissertation, following recommendations by SMUCKER *et al.* (2007), we use Fisher's randomization test, which is a non-parametric statistical significance test that does not make any assumptions regarding the underlying distribution of the scores produced by the retrieval system.

For Fisher's randomization test, the null hypothesis is that the runs labeled by system A and system B are identical and thus system A has no effect compared to system B . Under the null hypothesis, any permutation of the labels A and B is an equally likely output, and we can measure the difference between A and B for each permutation of the labels.

Given N queries, we could measure the number of permutations for which the difference in the retrieval metric \mathcal{M} was greater or equal to the actual difference between the systems. This number, divided by 2^{N+1} would be the exact two-sided p-value α . Computing 2^{N+1} permutations is not practical for large enough N 's (in our collections, $100 \leq N \leq 250$). Therefore, for efficiency reasons, we limit the number of permutations to 10,000 in our experiments. If $\alpha < 0.05$, we conclude that there is a statistically significant difference between the candidate retrieval system A and the baseline retrieval system B .

4.3 Summary

In this chapter, we described the standard TREC collections which include a document collection, a set of topics and a set of corresponding relevance judgments. In particular, we described the *Robust04*, *Gov2* and *ClueWeb-B* TREC collections, which are used in our experimental evaluation. In addition, we presented the binary and the graded relevance metrics which are commonly used in information retrieval research. Finally, we introduced Fisher's randomization test, a statistical significance test that is used to distinguish between the performance of the retrieval systems throughout this dissertation.

In the following chapters of this dissertation, we will evaluate the empirical results of our work using these TREC collections and evaluation criteria.

CHAPTER 5

PARAMETERIZED CONCEPT WEIGHTING

5.1 Introduction

In this chapter¹, we focus on the parameterized concept weighting in query hypergraphs. As described in Section 2.3, recently researchers found that employing supervised concept weighting is beneficial, especially for verbose natural queries. The supervised weighting techniques tend to outperform traditional unsupervised weighting methods that are based solely on inverse document frequency or inverse collection frequency weights (BENDERSKY and CROFT 2008; LEASE *et al.* 2009; ZHAO and CALLAN 2010).

Accordingly, in this chapter, we introduce a novel *weighted sequential dependence* (WSD) model. The WSD model is a weighted extension of a sequential dependence (SD) variant of a Markov random field model for information retrieval, first proposed by METZLER and CROFT (2005). It can also be viewed as a special case of a query hypergraph that incorporates parameterized concept weighting but does not employ dependencies between query concepts.

Unlike the previously proposed supervised concept weighting methods (BENDERSKY and CROFT 2008; LEASE *et al.* 2009; ZHAO and CALLAN 2010), the WSD method described in this chapter provides a generic framework for learning the importance of query concepts in a way that *directly* optimizes an underlying retrieval metric. The WSD method directly incorporates the concept weighting into the ranking function,

¹This chapter is partly based on the work published at the Third ACM International Conference on Web Search and Data Mining (BENDERSKY *et al.* 2010).

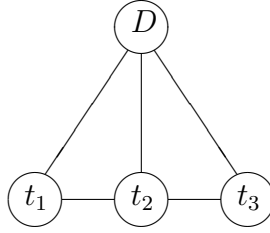


Figure 5.1. A Markov random field model for a three-term query under the sequential dependence assumption.

eliminating the need for a separate round of learning. In this manner, a metric divergence – which is often inherent to the other methods that combine query representation and ranking – is avoided. As we will show, this direct optimization strategy yields strong retrieval effectiveness gains.

The remainder of this chapter is organized as follows. First, in Section 5.2 we present a brief, self-contained overview of the Markov random field model and its sequential dependence model variant. Then, in Section 5.3, we present the weighted variant of the sequential dependence model and show that it can be modeled using a query hypergraph. In Section 5.4 we present an empirical evaluation of the weighted sequential dependence model using both TREC corpora and a proprietary web corpus. We conclude the chapter in Section 5.5.

5.2 Markov Random Field for Information Retrieval

A Markov random field (MRF) is an undirected graphical model that defines a joint probability distribution over a set of random variables. A Markov random field is defined by a graph G , where the nodes in the graph represent random variables and the edges define the dependence semantics between the random variables. In the context of information retrieval, the Markov random field models the joint distribution over a document random variable D and query term random variables t_1, \dots, t_N (denoted Q).

An example MRF for a three-term query is shown in Figure 5.1. In the MRF depicted in Figure 5.1, the adjacent query terms (e.g., t_1 and t_2) are dependent on each other since they share an edge, but non-adjacent query terms (e.g., t_1 and t_3) are independent given D .

The joint distribution over the document and query terms is generally defined as:

$$P_{G,\Lambda}(Q, D) = \frac{1}{Z_\Lambda} \prod_{c \in \text{Cliques}(G)} \psi(c; \lambda_c) \quad (5.1)$$

where $\text{Cliques}(G)$ is the set of cliques in G , each $\psi(c; \lambda_c)$ is a non-negative potential function defined over clique configuration c that measures the ‘compatibility’ of the configuration, Λ is a set of parameters that are used within the potential functions, and Z_Λ normalizes the distribution.

Therefore, to instantiate the MRF model, one must define a graph structure and a set of potential functions. METZLER and CROFT (2005) propose three different graph structures that make different dependence assumptions about the query terms. The *full independence* variant places no edges between query terms, the *sequential dependence* variant places edges between adjacent query terms (see Figure 5.1), and the *full dependence* variant places edges between all pairs of query terms. In this dissertation, we focus on the sequential dependence (SD) variant of the Markov random field, as it has been shown to provide a good balance between effectiveness and efficiency (METZLER and CROFT 2005).

Under the sequential dependence assumption, there are two types of cliques that we are interested in defining potential functions over. First, there are cliques involving a single term node and the document node. The potentials for these cliques are defined as follows:

$$\psi(q_i, D; \Lambda) = \exp\left(\lambda_{\text{QT}} f(t_i, D)\right).$$

It is common practice for MRF potential functions to have this type of exponential form, since potentials, by definition, must be non-negative. Here, $f(t_i, D)$ is a match-

ing function defined over the query term t_i and the document D , and λ_{QT} is a free parameter. The subscript QT denotes that these potentials are defined over the query terms.

The other cliques that we are interested in are those that contain two (adjacent) query term nodes and the document node. The potentials over these cliques are defined as:

$$\psi(t_i, t_{i+1}; \Lambda) = \exp \left(\lambda_{\text{PH}} f(\text{PH}(t_i, t_{i+1}), D) + \lambda_{\text{PR}} f(\text{PR}(t_i, t_{i+1}), D) \right)$$

where $f(\text{PH}(t_i, t_{i+1}), D)$ and $f(\text{PR}(t_i, t_{i+1}), D)$ are matching functions and λ_{PH} and λ_{PR} are free parameters. These potentials are made up of two distinct components. The first considers ordered (i.e., exact phrase) matches and is denoted by the PH subscript. The second, denoted by the PR subscript, considers proximity matches (refer back to Section 3.3.1 for the detailed definitions of these types of matches).

The matching function $f(\kappa, D)$ that is used by the Markov random field for information retrieval is identical to the concept matching function used by the query hypergraphs (see Equation 3.3) and is defined as

$$f(\kappa, D) \triangleq \log \frac{tf(\kappa, D) + \mu \frac{tf(\kappa, \mathcal{C})}{|\mathcal{C}|}}{\mu + |D|},$$

where κ can either be a query term t , an exact phrase $\text{PH}(t_i, t_{i+1})$, or a proximity match $\text{PR}(t_i, t_{i+1})$ (METZLER and CROFT 2005). For the detailed explanation of the components of this matching function, refer to Section 3.3.3.

After making the sequential dependence assumption and substituting the potentials $\psi(t_i, D; \Lambda)$, $\psi(t_i, t_{i+1}, D; \Lambda)$ into Equation 5.1, documents can be ranked according to:

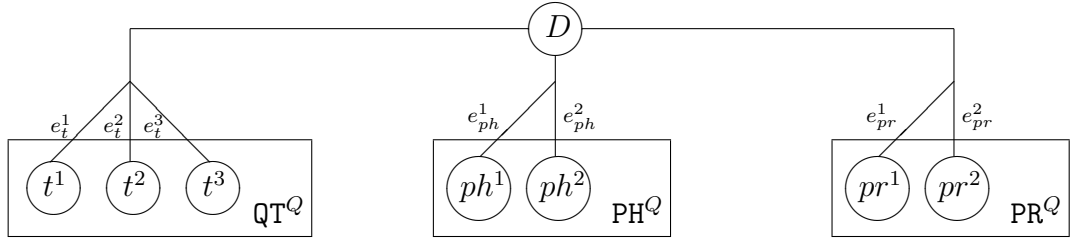


Figure 5.2. A hypergraph H^{SD} that encodes the sequential dependence model for a three-term query.

$$\begin{aligned}
 P(D|Q) \stackrel{\text{rank}}{=} & \lambda_{\text{QT}} \sum_{t_i \in Q} f(t_i, D) + \\
 & \lambda_{\text{PH}} \sum_{t_i, t_{i+1} \in Q} f(\text{PH}(t_i, t_{i+1}), D) + \\
 & \lambda_{\text{PR}} \sum_{t_i, t_{i+1} \in Q} f(\text{PR}(t_i, t_{i+1}), D) \quad (5.2)
 \end{aligned}$$

Conceptually, this ranking function is a weighted combination of a bag-of-words score, an exact bigram match score, and a proximity bigram match score. In this dissertation, we refer to the ranking function in Equation 5.2 as the *sequential dependence model* (SD). It has been shown that the parameters $\lambda_{\text{QT}} = 0.8$, $\lambda_{\text{PH}} = 0.1$, $\lambda_{\text{PR}} = 0.1$ are very robust and are optimal or near-optimal across a wide range of retrieval tasks (METZLER and CROFT 2005; METZLER and CROFT 2007b). Therefore, we use this parameter setting in the remainder of this dissertation.

5.3 Weighted Sequential Dependence Model

Note that the Markov random field model, as defined by METZLER and CROFT (2005), is a special case of a query hypergraph. The *cliques* and the *potentials* in the Markov random field model are mapped to the *concepts* and the *factors* in the query hypergraph, respectively.

For instance, the sequential dependence variant of the MRF, described in Section 5.2, can be easily represented using a query hypergraph H^{SD} presented in Figure 5.2. The hypergraph H^{SD} is constructed as follows:

- (a) H^{SD} contains three structures, query terms (QT), bigram phrases (PH) and bigram proximity matches (PR). Formally, $\Sigma^Q \triangleq \{\text{QT}, \text{PH}, \text{PR}\}$.
- (b) H^{SD} contains only local edges that are associated with local factors $\phi(\{\kappa\}, D)$, as defined by Equation 3.4.
- (c) H^{SD} is parameterized by structure. Formally,

$$\forall \kappa_i, \kappa_j \in \sigma : \lambda(\kappa_i) = \lambda(\kappa_j) = \lambda(\sigma),$$

where $\sigma \in \Sigma^Q$.

There are two important advantages, however, to representing queries using hypergraphs, as opposed to query representation using the MRF model as defined by METZLER and CROFT (2005). First, query hypergraphs are defined over arbitrary concepts rather than single query terms. This allows the hypergraphs to model the dependencies between arbitrary concepts rather than terms.

Second, the query hypergraphs can be parameterized by concept, rather than by structure, which allows for a more fine-grained weighting of query concepts, which can be especially beneficial for verbose queries. In this section, we focus on this second advantage and demonstrate how the SD model can be extended into a *weighted sequential dependence* (WSD) model using a query hypergraph representation.

First, we can express the SD ranking function in Equation 5.2 using a notation for the query hypergraph H^{SD} (as defined above) as

$$s_{\text{SD}}(Q, D) \triangleq \sum_{\sigma \in \Sigma^Q} \lambda(\sigma) \sum_{\kappa \in \sigma} f(\kappa, D).$$

<i><title> american indian museum</i>	
Terms	Phrases
.502 american	.166 american indian
.557 indian	.166 indian museum
.592 museum	
<i><desc> “What are the plans for a national museum of the American Indian?”</i>	
Terms	Phrases
.051 plans	.022 plans national
.092 national	.062 national museum
.119 museum	.022 museum american
.101 american	.051 american indian
.112 indian	

Figure 5.3. Examples of weighted *<title>* and *<desc>* queries for TREC topic §664. Common stopwords are automatically removed from the queries prior to weight assignment.

To go beyond the parameterization by structure, we will parameterize the weights $\lambda(\cdot)$ based on the concepts themselves rather than their respective structures. Since assigning a single weight for each concept in the vocabulary is infeasible, we employ the parameterization-by-concept technique described in Section 3.4.2. Recall, that in this approach we parameterize each concept using a combination of importance features Φ . These features include frequencies both from the collection itself and from external sources. The set of features is detailed in Table 3.2.

Using the parameterization-by-concept approach, we can define

$$\forall \kappa \in \sigma: \lambda(\kappa) = \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \varphi(\kappa).$$

We can now substitute the structure weight $\lambda(\sigma)$ for the above definition of $\lambda(\kappa)$ in the SD model ranking function, which yields

$$\begin{aligned}
sc_{\text{WSD}}(Q, D) &\triangleq \sum_{\sigma \in \Sigma^Q} \sum_{\kappa \in \sigma} \lambda(\kappa) f(\kappa, D) \\
&= \sum_{\sigma \in \Sigma^Q} \sum_{\kappa \in \sigma} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \varphi(\kappa) f(\kappa, D) \\
&= \sum_{\sigma \in \Sigma^Q} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) \tag{5.3}
\end{aligned}$$

The resulting scoring function in Equation 5.3 is reminiscent of the general hypergraph ranking function in Equation 3.7, if the global factor component would be dropped. Therefore, the WSD ranking function takes into account the individual concept weights, but not the dependencies between them.

Note that the ranking function in Equation 5.3 is linear in the set of free parameters $\Lambda = \lambda(\varphi, \Sigma)$. Therefore, we can directly use the coordinate ascent algorithm for parameter optimization (see Figure 3.3). The number of the free parameters in this function, which we call *weighted sequential dependence* (WSD) model, is

$$|\Lambda| = |\Sigma^Q| |\Phi| = 3 \cdot 6 = 18.$$

It is important to note here the major difference between the WSD method and some previously proposed methods for query concept weighting (BENDERSKY and CROFT 2008; LEASE *et al.* 2009; ZHAO and CALLAN 2010) and query segmentation (BERGSMA and WANG 2007; BENDERSKY *et al.* 2009; TAN and PENG 2008). The proposed WSD method provides a generic framework for learning the importance of query term concepts in a way that *directly* optimizes an underlying retrieval metric. This is different from the previous methods that learn query concept weighting and query segmentation based on a surrogate metric, e.g., the probability of the concept given a set of relevant documents (ZHAO and CALLAN 2010) or the segmentation accuracy (BERGSMA and WANG 2007).

In other words, unlike these previously proposed methods, the WSD method directly incorporates the concept weighting into the ranking function, avoiding the need for a

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
QL	34.86	24.43	50.41	29.56	29.27	18.48
SD	36.20	25.85*	53.82*	30.90*	31.04	19.37
WSD	36.47*	26.09*	54.09*	31.68*_†	31.25	20.23*_†

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
QL	33.09	24.24	47.62	25.66	23.85	12.75
SD	35.04*	25.62*	51.11*	27.97	22.97*	12.99
WSD	37.05*_†	27.41*_†	52.21*	29.36*_†	25.31_†	14.56*_†

Table 5.1. Retrieval evaluation based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the QL and the SD methods are marked by * and †, respectively.

separate round of learning. In this manner, we avoid the issue of metric divergence that is often inherent to the other methods that combine query representation and ranking. As we will show, this strategy yields strong retrieval effectiveness gains.

Figure 5.3 shows an example of the weighted $\langle title \rangle$ and $\langle desc \rangle$ queries for the TREC topic §664 when the WSD method is applied. As can be seen from Figure 5.3, the weighting of the verbose $\langle desc \rangle$ query assigns higher weights to the terms that appear in the $\langle title \rangle$ query *american indian museum*. This demonstrates the ability of the WSD method to correctly upweight the key query terms. In addition, the key phrases *american indian* and *national museum* are assigned the highest weights in the verbose $\langle desc \rangle$ query.

5.4 Evaluation

5.4.1 Evaluation on TREC corpora

We compare the performance of our weighted sequential dependence model (WSD) to two baseline retrieval models. The first is the query-likelihood model (QL) (PONTE and CROFT 1998), a standard bag-of-words retrieval model implemented in the Indri search engine. The second is the unweighted sequential dependence model (SD) as

described in Section 5.2. All the initial retrieval parameters are set to the default Indri values, which reflect the best-practice settings. All the training and evaluation is done using 3-fold cross-validation. The statistical significance of the differences in the performance of the retrieval methods is determined using a Fisher’s randomized test with 10,000 iterations and $\alpha < 0.05$.

We measure the performance using standard retrieval metrics for TREC corpora, as described in Section 4.2. For metrics that use binary relevance judgments, we use precision at the top 20 retrieved documents ($P@20$) and mean average precision across all the queries (MAP). For metrics that use graded relevance judgments, we use normalized discounted cumulative gain and expected reciprocal rank at rank 20 ($NDCG@20$ and $ERR@20$, respectively). We evaluate the retrieval methods under comparison using the three TREC corpora shown in Table 4.1.

When estimating the parameters for the WSD model using coordinate ascent, we use mean average precision as the target evaluation metric \mathcal{M} (see Figure 3.3 for more details). This is due to the fact that MAP is known to be a stable measure (BUCKLEY and VOORHEES 2004), as it measures the quality of the entire ranked list.

In our evaluation we use both the $\langle title \rangle$ and the $\langle desc \rangle$ portions of TREC topics as queries. As described in Section 4.1, $\langle title \rangle$ queries are generally short, and can be viewed as keyword queries on the topic. $\langle desc \rangle$ queries are generally more verbose and syntactically richer natural language expressions of the topic. For instance, the queries in Figure 5.3 are examples of $\langle title \rangle$ and $\langle desc \rangle$ queries on the same topic, respectively.

Table 5.1 shows the summary of the binary retrieval metrics for the three TREC corpora for both $\langle title \rangle$ and $\langle desc \rangle$ queries. It is evident that both sequential dependence models (SD and WSD) outperform the query likelihood model (QL) in almost all the cases on all the metrics. This verifies the positive impact of the inclusion of phrases and proximities into the query representation on the retrieval performance.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
QL	11.22	40.14	16.47	40.90	8.40	19.75
SD	11.69	41.78*	17.09	43.23*	8.80	21.36*
WSD	11.68*	42.02*	17.34	44.06*	9.41*	22.20*

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
QL	11.44	38.75	15.06	37.89	7.32	17.74
SD	11.76	40.91*	15.73	40.97*	7.58	17.11
WSD	12.04*	42.86*_†	16.52*_†	42.47*_†	8.58*	19.58_†

Table 5.2. Retrieval evaluation based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the QL and the SD methods are marked by * and †, respectively.

For the two sequential dependence models, the weighted sequential dependence model (WSD) outperforms the unweighted one (SD) on all collections in terms of *MAP* (which is used as our metric for direct optimization). The largest gains in *MAP* can be seen for the verbose $\langle desc \rangle$ queries, where there is always a statistically significant difference between the WSD and the SD models (in terms of *MAP*).

It is interesting to note that even for the *P@20* metric, which is not directly optimized, WSD is more effective than SD in all comparisons. This validates the effectiveness and the robustness of the coordinate ascent optimization using mean average precision as a target metric.

Table 5.2 shows the summary of the graded retrieval metrics for the three TREC corpora for both $\langle title \rangle$ and $\langle desc \rangle$ queries. The evaluation for the graded metrics is in line with the evaluation using the binary retrieval metrics. The WSD method is the best among the evaluated methods in all but one comparison. The gains attained by the weighted sequential dependence model are the largest for the verbose $\langle desc \rangle$ queries: WSD method is statistically significantly better than the SD method (in terms of *NDCG@20*) for all the TREC corpora.

	% queries (50+% gain)	% queries (50+% loss)	% gain
$\langle title \rangle$	3.67	0.96	2.6
$\langle desc \rangle$	18.64	3.04	8.1

Table 5.3. Average effect of concept weighting method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the *MAP* metric).

It is also interesting to examine the relative gains from using the weighted variant of the sequential dependence model (compared to its unweighted variant) across all corpora for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Recall that we hypothesized that while concept weighting is important for all queries, it benefits the longer, more verbose queries to a larger degree due to the fact that they tend to include concepts that have varying importance for expressing the query intent.

For instance, consider the queries in Figure 5.3. All the concepts in the $\langle title \rangle$ query in Figure 5.3 are key concepts for expressing the query intent, and are assigned roughly the same weights by the WSD method. On the other hand, the $\langle desc \rangle$ query has much more weight variance. For instance, the term *indian* is deemed twice as important as the term *plans* by the WSD method.

Table 5.3 examines the difference in effectiveness gains (as measured by *MAP*) as a result of applying the WSD method to both $\langle title \rangle$ and $\langle desc \rangle$ queries averaged across the three corpora. Table 5.3 clearly demonstrates that while concept weighting is beneficial for both types of queries, its effect is much more pronounced for the verbose $\langle desc \rangle$ queries. While it significantly hurts slightly more $\langle desc \rangle$ queries than $\langle title \rangle$ queries (3.04% vs. 0.96%, respectively), it has a significant positive impact (more than 50% effectiveness gain) on almost 19% of $\langle desc \rangle$ queries, compared to less than 4% of the $\langle title \rangle$ queries. In addition, the overall average effectiveness gain as a result of concept weighting is more than three times higher for the $\langle desc \rangle$ queries.

5.4.2 Evaluation on a commercial web corpus

As shown in the previous section, the weighted variant of the sequential dependence model demonstrates significant retrieval effectiveness improvements on three TREC collections. In this section, we describe a set of experiments that explores whether these gains can be directly transferred into a web search setting. To this end, we test the ranking with a weighted sequential dependence model on a proprietary web corpus provided by a large commercial search engine.

The experiments with this proprietary web corpus were performed while the author was on a summer internship at Yahoo! Research. These experiments were also published by BENDERSKY *et al.* (2010)

Since graded relevance metrics are the most common way to evaluate web search engines (BURGES *et al.* 2005; CHAPELLE *et al.* 2009), we only report these metrics. In particular, we report the *non-normalized* discounted cumulative gain at ranks 1 and 5 ($DCG@1$ and $DCG@5$, respectively). However, in the optimization of the weighted sequential dependence model parameters, we use the *total* discounted cumulative gain – i.e. the discounted cumulative gain at the total depth of the ranked list – as the target metric \mathcal{M} .

Similarly to the TREC experiments, during the development phase, we found that the results attained by optimizing this metric were more stable over all ranks than the results attained by optimizing for the discounted cumulative gain at a particular rank. This can be attributed to the fact that the total discounted cumulative gain incorporates information about the entire ranked list, whereas $DCG@1$ and $DCG@5$ only consider the top ranked documents and are more prone to bias and overfitting.

To differentiate between the effect of concept weighting on queries of varying length, as was done in the case of TREC corpora, we divide the queries into three groups based on their length. Length is defined as a number of word tokens separated by space in the query.

The first group of queries (*Len-2*) includes very short queries of length two. The second group (*Len-3*) includes queries of length three. The third group (*Len-4+*) consists of more verbose queries of length varying between four and twelve.

While the queries in the first two groups mostly have a navigational intent, the queries in the third group tend to be more complex informational queries. For each group, we randomly sample a 1,000 web search queries for which relevance judgments are available. We then train and evaluate (using five fold cross-validation) a separate sequential dependence model and weighted sequential dependence model for each group.

Table 5.4 shows the summary of the retrieval results on the three query groups. Table 5.4 demonstrates two important findings. First, including term dependence information is highly beneficial for queries of all lengths. SD attains up to 15.4% improvement over QL, which is a bag-of-words model. This result is highly significant, given the large size of our query set.

Second, concept weighting results in significant improvements for longer (*Len-4+*) queries, and its performance is comparable for shorter queries to the performance of the unweighted dependence model (slight improvement on *Len-2* and slight decrease in performance on *Len-3*). For group *Len-4+*, WSD attains improvement of close to 2.5% for *DCG@5*. This is a highly significant improvement, especially when taking into account the importance of relevance at top ranks for the web search task.

These results using a proprietary web corpus further demonstrate the importance of concept weighting for verbose search queries. For both TREC and web corpora, WSD is significantly more effective than SD for this type of queries.

5.5 Summary

In this chapter, we focused on the parameterized concept weighting in query hypergraphs. As a result, we introduced a novel weighted sequential dependence (WSD)

	Len-2		Len-3		Len-4+	
	<i>DCG@1</i>	<i>DCG@5</i>	<i>DCG@1</i>	<i>DCG@5</i>	<i>DCG@1</i>	<i>DCG@5</i>
QL	0.803	2.231	0.784	2.290	0.629	1.691
SD	0.926	2.733	1.008	2.971	0.864	2.383
WSD	0.929	2.754	0.995	2.929	0.884	2.443

- All the differences are statistically significant

Table 5.4. Comparison of retrieval results over a sample of web queries with query likelihood (QL), sequential dependence model (SD) and the weighted sequential dependence model (WSD). Discounted cumulative gain at ranks 1 and 5 is reported.

model. The WSD model is a weighted extension of a sequential dependence (SD) variant of a Markov random field model for information retrieval METZLER and CROFT (2005). Weighted sequential dependence model can also be viewed as a special case of a query hypergraph that incorporates parameterized concept weighting but does not employ dependencies between query concepts.

In Section 5.2 we presented a brief, self-contained overview of the Markov random field model. Then, in Section 5.3, we presented the weighted variant of the sequential dependence model and showed that it can be modeled using a query hypergraph. In Section 5.4 we presented an empirical evaluation of the weighted sequential dependence model using both TREC corpora and a proprietary web corpus. This empirical evaluation demonstrates the retrieval effectiveness of the WSD model, especially for verbose queries.

After presenting the parameterized concept weighting of query concepts in this chapter, in the next two chapters we focus on parameterized query expansion using either the retrieval corpus (Chapter 6) or multiple external information sources (Chapter 7). In both cases, we adopt the parameterized concept weighting approach developed in this chapter to assign weights to expansion concepts that do not explicitly occur in the original search query.

CHAPTER 6

PARAMETERIZED QUERY EXPANSION

6.1 Introduction

The main shortcoming of the weighted sequential dependence model presented in the previous chapter, is that the weighting is performed exclusively on the concepts that explicitly occur within the query and disregards the *expansion* concepts associated with the information need underlying the query (e.g., the concepts distilled by state-of-the-art query expansion approaches such as relevance model (LAVRENKO and CROFT 2003) or latent concept expansion (METZLER and CROFT 2007a)). Accordingly, in this chapter, we explore the question of how to seamlessly and effectively integrate these expansion concepts within a query representation that supports parameterized concept weighting such as query hypergraphs.

To address this question, in this chapter¹, we propose a novel *parameterized query expansion* model. The proposed model provides an effective alternative to the standard unsupervised weighting for both single terms and multiple-term concepts, similarly to the weighted sequential dependence model described in the previous chapter. In addition, the model generalizes the current supervised concept weighting approaches (BENDERSKY *et al.* 2010; LEASE 2009; SHI and NIE 2010; SVORE *et al.* 2010; WANG *et al.* 2010) and provides a *unified* framework for weighting both explicit and explicit query concepts.

¹This chapter is partly based on the work published at the 34th Annual ACM SIGIR Conference (BENDERSKY *et al.* 2011).

Query Terms	Query Bigrams	Expansion Terms
.1064 patrol	.0257 civil air	.0639 cadet
.1058 civil	.0236 air patrol	.0321 force
.1046 training	.0104 training participants	.0296 aerospace
.0758 participants	.0104 participants receive	.0280 cap

Table 6.1. Explicit and expansion concepts with the highest importance weight for the query “*What is the current role of the civil air patrol and what training do participants receive?*”.

As an illustrative example of the parameterized query expansion in action, consider the verbose query

“What is the current role of the civil air patrol and what training do participants receive?”

Table 6.1 shows the most important explicit query concepts (terms and bigram phrases) and the most important expansion terms learned by our model. Note that the weights assigned by our model are different from the weights that would be assigned by inverse document frequency (IDF) weight alone. For instance, while the term *air* has higher IDF than the term *training*, it is deemed less important for the query. In addition, while the term *air* is not important on its own, it is significant in the context of the bigram *air patrol*.

In the case of the query in Table 6.1, the parameterized query expansion model improves the retrieval effectiveness by 64% over the standard query-likelihood model (QL) (PONTE and CROFT 1998), by 21% over the WSD model described in the previous section, and by 8% over the latent concept expansion model (METZLER and CROFT 2007a). As the evaluation in Section 6.5 demonstrates, these gains in retrieval effectiveness are consistent across queries and collections.

Expanding the query with related term or concepts has a long history in information retrieval (ROCCHIO, J. 1971; XU and CROFT 1996; LAVRENKO and CROFT 2003; METZLER and CROFT 2007a). One technique that is commonly used for query

expansion is pseudo-relevance feedback. Pseudo-relevance feedback allows the system to leverage information from the underlying retrieval corpus in order to expand the query with related terms or concepts without requiring an explicit user interaction. This is also an approach we adopt in this dissertation.

While there is a large number of successful pseudo-relevance feedback based retrieval models (e.g., (CAO *et al.* 2008; LAVRENKO and CROFT 2003; METZLER and CROFT 2007a; LV and ZHAI 2010; XU and CROFT 1996)), most of them employ unsupervised weighting for both explicit and expansion concepts. A notable exception is the work by CAO *et al.* (2008) which uses binary classification to determine the importance of the expansion terms. Unlike CAO *et al.* (2008), the proposed parameterized query expansion method takes a more holistic approach, and assigns importance weights to *both* explicit and expansion concepts.

The remainder of this chapter is organized as follows. First, in Section 6.2, we outline the theoretical foundations of pseudo-relevance feedback and the state-of-the-art latent concept expansion model (METZLER and CROFT 2007a). Then, in Section 6.3, we describe the process of parameterized query expansion with query hypergraphs. In Section 6.4 we specify the parameter optimization in the parameterized query expansion model. In Section 6.5 we empirically evaluate the performance of the parameterized query expansion model. We conclude the chapter in Section 6.6.

6.2 Pseudo-Relevance Feedback

Query expansion using related terms or concepts has a long history of success in information retrieval. One approach commonly used for automatic query expansion is the pseudo-relevance (PRF). In the PRF approach, the underlying retrieval corpus is leveraged to automatically expand the query with related terms that can improve the retrieval effectiveness of the original query.

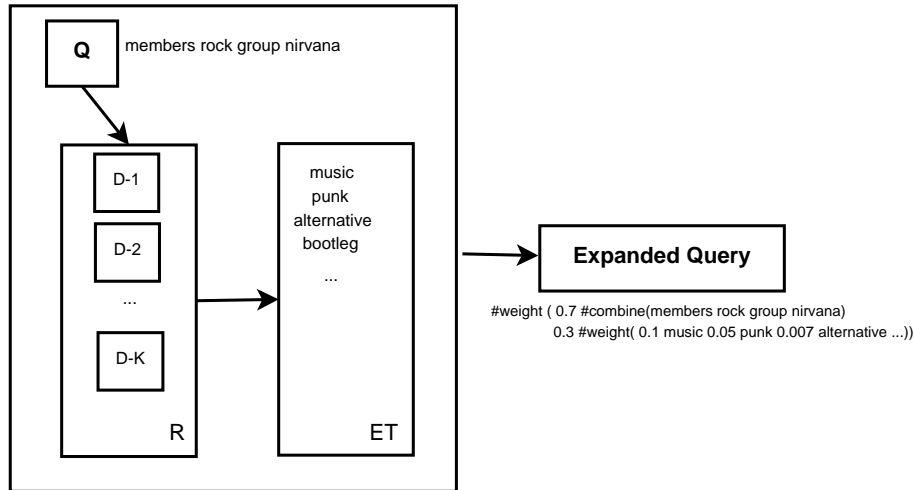


Figure 6.1. Schematic diagram of query expansion using pseudo-relevance feedback from the retrieval corpus.

The pseudo-relevance feedback approach automates the process of relevance feedback by forgoing the need for the user of the retrieval system to indicate a set of true relevant documents. In fact, previous research shows that PRF can often enable improvements in retrieval effectiveness without requiring any extra interaction from the user.

Figure 6.1 shows a schematic diagram of the pseudo-relevance feedback process. First, a query Q is issued to the retrieval corpus, and a first round of retrieval is performed. A set of documents retrieved at the top K positions (denoted \mathcal{R}) is referred to as the *pseudo-relevant* set. This is due to the fact that the true relevant set of documents for a given query Q is unknown a priori. Therefore, this true relevant set is approximated using the highest ranked documents in response to the query Q .

The pseudo-relevant set \mathcal{R} is then used for extracting a list of terms or concepts that are related to the original query. There are various methods for extracting this list of terms or concepts that are related to the query, some of which are discussed next. Once this list is obtained, the query is expanded with the extracted terms or

concepts and issued again to the search engine for the final round of retrieval, the results of which are presented to the user.

Most often, the expanded query takes a weighted form, similarly to the example Indri query shown in Figure 6.1, which combines the original query “*members rock group nirvana*” with expansion terms *music*, *punk*, *alternative*, etc. The original and the expanded query parts are assigned importance weights. In addition, each of the expansion terms or concepts is assigned a weight based on the strength of its relatedness to the information need expressed by the original query. The various PRF methods differ in the assignment of these concept weights.

There is an abundance of literature on query expansion using pseudo-relevance feedback. One of the most successful of these expansion methods is the relevance model proposed by LAVRENKO and CROFT (2003). In this model, the expansion term weight is determined by its probability of being generated by a *relevance model*, which is approximated by the pseudo-relevant set \mathcal{R} . Formally,

$$w_{\text{RM}}(t) \triangleq P(t|\mathcal{R}) \approx \sum_{D \in \mathcal{R}} P(t|D) \prod_{q_i \in Q} P(q_i|D).$$

Note that this formulation of the relevance model is, in fact, a bag-of-words approach, since it assumes independence between the query terms and the expansion term t .

When we define the probabilities $P(\cdot|D)$ in the equation above as maximum likelihood estimates with Dirichlet smoothing, the weight of the expansion term t in the relevance model can be expressed using the definition of the matching function f in Equation 3.3. Accordingly, we can rewrite the equation above as

$$w_{\text{RM}}(t) \triangleq \sum_{D \in \mathcal{R}} \exp \left(\sum_{q_i \in Q} f(q_i, D) + f(t, D) \right).$$

After its initial introduction by LAVRENKO and CROFT (2003), the relevance model was further expanded and generalized by other researchers to incorporate,

among other things, more complex weighting schemes (CAO *et al.* 2008), term proximities (LV and ZHAI 2010), and random walks over expansion term graphs (COLLINS-THOMPSON and CALLAN 2005). One of the most important and empirically successful generalizations of the relevance model called latent concept expansion (LCE) was recently proposed by METZLER and CROFT (2007a). Latent concept expansion has several important advantages, including state-of-the art retrieval performance (METZLER and CROFT 2007a; LANG *et al.* 2010) and the ability to leverage information about arbitrary query concepts to improve the quality of query expansion.

To obtain the list of expansion concepts using LCE, one need not make any assumptions about the independence between the concepts in the query and the expansion concepts. Instead, we assume the existence of an arbitrary scoring function $sc(Q, D)$ that assigns a relevance score to a document D in response to the query Q . Then, the weight of the expansion concept κ is calculated using

$$w_{\text{LCE}}(\kappa) = \sum_{D \in \mathcal{R}} \exp \left(\gamma_1 sc(Q, D) + \gamma_2 f(\kappa, D) - \gamma_3 \log \frac{tf_{\kappa, \mathcal{C}}}{|\mathcal{C}|} \right), \quad (6.1)$$

where γ_i 's are free parameters.

As evident from Equation 6.1, w_{LCE} combines three key features to assign a weight to concept κ :

- (a) The relevance of all the pseudo-relevant documents $D \in \mathcal{R}$, which contain the expansion concept κ – as manifested by the document score $sc(Q, D)$.
- (b) The impact of the match of the expansion concept κ in the pseudo-relevant documents – expressed by the matching function $f(\kappa, D)$.
- (c) The inverse collection frequency (ICF) of the concept κ , which is calculated by the factor $-\log \frac{tf_{\kappa, \mathcal{C}}}{|\mathcal{C}|}$. The ICF factor dampens the weights of very common words, thereby reducing the number of non-content-bearing concepts in the expansion list.

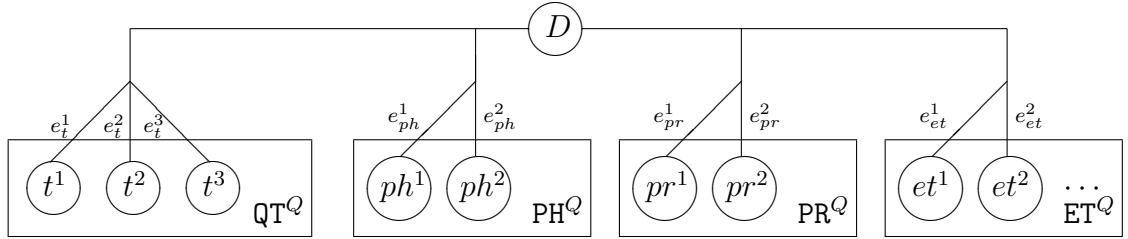


Figure 6.2. A hypergraph H^{PQE} that encodes the parameterized query expansion model for a three-term query.

Latent concept expansion can be adopted to include any arbitrary concept type for query expansion. However, in this dissertation we limit the expansion to individual terms. First, this focus improves the overall efficiency of the query expansion. Second, previous work found no significant benefits when additional types of latent concepts (such as phrases) were associated with the query in addition to terms alone (METZLER and CROFT 2007a).

The LCE approach is general enough to incorporate multiple types of scoring and matching functions to weight an expansion concept. However, it still lacks the flexibility of the fully parameterized concept weighting model (introduced in Chapter 5) that allows the use of an arbitrary set of concept importance features for concept weighting. In the next section, we show that the LCE approach can be further generalized by using query hypergraphs, which incorporate the concept importance features in the expansion concept weighting. We refer to this approach as *parameterized query expansion*.

6.3 Parameterized Query Expansion with Query Hypergraphs

In this section, we introduce the *parameterized query expansion* (PQE) approach that enables to perform query expansion using the query hypergraph representation. Recall from Section 3.1 that the concepts modeled by the query hypergraph H are not limited to the concepts that explicitly occur in the original user query. Instead,

any concept that is related to the information need expressed by the query can be added to the query hypegraph H as a vertex.

In this manner, query hypergraphs provide a flexible framework for performing query expansion. As Figure 6.2 shows, query expansion can be straightforwardly modeled by integrating an additional *expansion terms structure*, denoted ET, into the query hypegraph. This structure contains the expansion terms that are associated with the original query, e.g. the terms that were obtained through the process of pseudo-relevance feedback.

As stated in the previous section, we limit our attention to expansion using single terms rather than arbitrary concepts. This restriction is mainly due to the efficiency considerations, since query latency is an important concern in information retrieval applications. However, from the purely theoretical perspective, query hypergraphs can also incorporate arbitrary expansion concepts rather than single terms.

Any of the techniques described in Section 6.2 can be applied for obtaining the set of expansion terms in the ET structure. For instance, we could use the bag-of-words relevance model (LAVRENKO and CROFT 2003), or the latent concept expansion that better accounts for the dependencies between the query and the expansion terms (METZLER and CROFT 2007a).

Instead, in this section we explore a novel query expansion technique that leverages the parameterized concept weighting approach described in Chapter 5 for performing a more effective query expansion. Recall that the LCE approach uses a dampening ICF factor that reduces the weight of common expansion terms (see Equation 6.1). While ICF was shown to be a valuable factor for an effective expansion term weighting (METZLER and CROFT 2007a; LANG *et al.* 2010), it can be further enhanced by considering the fully parameterized approach.

Instead of a single dampening factor, let us associate each expansion term κ with a set of importance features Φ . For simplicity, the set Φ is identical to the feature set

used for assigning the weights to the explicit concepts in the query (see Table 3.2). Using the importance features in the set Φ , we can represent the expansion term weight using a parameterized concept weight

$$w_{\text{PCW}}(\kappa) \triangleq \sum_{\varphi \in \Phi} \lambda(\varphi, \text{ET}) \varphi(\kappa).$$

Further, recall that the importance weights are also used in assigning a relevance score to document D in response to query Q in a parameterized concept weighting approach. An example of such approach is the weighted sequential dependence model (WSD) presented in Equation 5.3. In this approach, we assign parameterized concept weights to query terms (represented by the **QT** structure), phrases (**PH** structure) and proximity matches (**PR** structure), and incorporate these weights in the ranking function

$$s_{\text{WSD}}(Q, D) = \sum_{\sigma \in \{\text{QT, PH, PR}\}} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D)$$

Therefore, when considering the weight assigned to an expansion term by a pseudo-relevance feedback based approach such as w_{LCE} in Equation 6.1, the parameterized concept weights play a dual role. First, via their inclusion in the ranking function, they determine the selection and the scores of the documents in the pseudo-relevant set \mathcal{R} . Second, they impact the weights of the expansion terms selected from the pseudo-relevant set.

Accordingly, we base the parameterized query expansion (**PQE**) approach on the general form of the **LCE** weighting presented in Equation 6.1. First, we substitute the ranking function in Equation 6.1 by the **WSD** ranking function $s_{\text{WSD}}(Q, D)$. Second, we substitute the **ICF** dampening factor by the general parameterized concept weight w_{PCW} . The resulting expansion concept weight is

$$\begin{aligned}
w_{\text{PQE}}(\kappa) &\triangleq \sum_{D \in \mathcal{R}} \exp \left(s_{C_{\text{WSD}}}(Q, D) + f(\kappa, D) + w_{\text{PCW}}(\kappa) \right) = \\
&= \sum_{D \in \mathcal{R}} \exp \left(\sum_{\sigma \in \{\text{QT,PH,PR}\}} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) + \right. \\
&\quad \left. + f(\kappa, D) + \sum_{\varphi \in \Phi} \lambda(\varphi, \text{ET}) \varphi(\kappa) \right). \tag{6.2}
\end{aligned}$$

Note that the free parameters in Equation 6.2 are now governed by the set of importance features Φ , rather than the fixed weights γ_i as in Equation 6.1. This change improves the expansion term selection in two ways:

- (a) The weight of the expansion term is increasing if it occurs in documents that contain many highly weighted explicit query concepts.
- (b) The weight of the expansion term varies based on the values of all the importance features associated with the term (and not just a single ICF factor).

Once we obtained a set of expansion terms, it is captured by the expansion term structure ET in the query hypergraph H . Then, to assign a relevance score to document D in response to query Q , we use the parameterized concept weighting approach, and use the weighted concept matches from both the explicit query concept and the expansion terms. Thus, the PQE ranking function is

$$s_{C_{\text{PQE}}}(Q, D) \triangleq \sum_{\sigma \in \{\text{QT,PH,PR,ET}\}} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) \tag{6.3}$$

To complete the derivation of the PQE retrieval model, in the next section we describe the pipeline optimization of the parameters $\lambda(\varphi, \sigma)$ in Equation 6.3.

6.4 Parameter Optimization

The weighted sequential dependence model (WSD), a parameterized concept weighting approach presented in Chapter 5 only considers the weighting of the concepts that

<i>camels in north america</i>	
LCE expansion terms	PQE expansion terms
indians	bison
mexico	oil
new	NAFTA
dress	fossil
clothing	expansion
...	...
AP = 0.07	AP = 0.49

Table 6.2. Examples of expansion terms obtained by the LCE and the PQE methods for the query “camels in north america”.

PQEOptimization(Λ^0)

- 1: $\Lambda_Q^0 = \Lambda_{\{\text{QT,PH,PR}\}}^0$
- 2: $\Lambda_E^0 = \Lambda_{\{\text{ET}\}}^0$
- 3: $\langle \mathcal{M}, \Lambda_Q \rangle \leftarrow \text{CoordinateAscent}(\emptyset, \Lambda_Q^0)$
- 4: $\langle \mathcal{M}, \Lambda_E \rangle \leftarrow \text{CoordinateAscent}(\Lambda_Q, \Lambda_E^0)$
- 5: **return** $\langle \mathcal{M}, \Lambda_Q \cup \Lambda_E \rangle$

Figure 6.3. Pipeline optimization of the parameterized query expansion method.

explicitly occur in the query. In contrast, the parameterized query expansion (PQE) combines weighting of the explicit query concepts with the weighting of the expansion terms obtained through pseudo-relevance feedback.

Since PQE combines evidence from both the explicit query and the ranked list produced by this query (refer to Figure 6.1 for the outline of the pseudo-relevance feedback process), the parameterization of the concepts that explicitly occur in the query (concepts in the structures QT, PH, and PR) will have a direct effect on the expansion terms that are included in the expansion terms structure ET.

As an example consider the expansion terms obtained by the LCE expansion approach METZLER and CROFT (2007a) and the PQE expansion approach for the query “*camels in north america*” presented in Table 6.2. The LCE expansion approach does not employ parameterized concept weighting in the expansion stage, while the PQE

expansion approach assigns weights to the explicit query concepts using the weighted sequential dependence model.

There is a stark difference between the two expansion term lists in Table 6.2. The LCE list focuses on terms related to the Native Americans, while the PQE list focuses on fossils and other North American animal species that went extinct. This difference results in a significant increase in average precision of the query (0.49 for the PQE approach, compared to the 0.07 for the LCE approach).

Motivated by this example, instead of using a single round of optimization of the free parameters Λ in the PQE ranking function in Equation 6.3, we propose a novel two-stage pipeline optimization technique. While simple, this two-stage technique is effective for learning robust weights for both explicit and latent query concepts, as well as improving the quality of the set of ET-concepts.

We base our approach on the general pipeline optimization algorithm first presented in Figure 3.4. The algorithm in Figure 6.3 provides a schematic overview of this two-stage pipeline optimization.

First, we denote the initial parameterization of the explicit query concepts (concepts in the QT, PH, and PR structures) Λ_Q^0 , and the initial parameterization of the expansion terms Λ_E^0 . At the first stage of the pipeline optimization algorithm (line 3 in Figure 6.3), we include only the explicit concept types $\{\text{QT, PH, PR}\}$ for optimizing the initial parametrization Λ_Q^0 . This process obtains an *optimized* parameterization Λ_Q , which is used for obtaining the pseudo-relevant set \mathcal{R} and a large pool of expansion terms to be included in the ET structure. We limit the size of this large pool to at most 100 terms in our experiments. As Table 6.2 illustrates, the expansion terms using the optimized parameterization Λ^Q , can be radically different from the one obtained using a non-parameterized retrieval model (as in the case of the LCE approach).

At the second stage of the training phase, we include both explicit query concepts and the expansion terms from the ET structure for optimizing the initial parameterization Λ_E^0 (line 4 in Figure 6.3). Note that the optimized parameterization of the explicit query concepts Λ_Q is kept fixed during this process.

This second round of the coordinate ascent algorithm may be computationally intensive, especially for the web-scale collections, since the query expansion produces queries that require a large number of concept matches in the ranked documents. To alleviate this problem to some degree, and to make the optimization process more efficient, at each iteration of the coordinate ascent algorithm, we include in the expanded queries at most 10 expansion terms with the highest weight (as determined by the parameterization Λ_E^i at the i -th iteration of the coordinate ascent algorithm) from the initial large expansion term pool of 100 terms.

The optimization phase concludes after this second round of the coordinate ascent algorithm is completed. At this point, the entire set of parameters Λ is optimized in terms of the target retrieval metric \mathcal{M} .

In this way, we ensure that the parameters Λ in the PQE ranking function (Equation 6.3) are optimized to deliver both the best selection of the expansion terms and the most effective retrieval performance of the expanded queries. As our experimental results demonstrate, this leads to a significant improvement over the state-of-the-art non-parameterized retrieval methods that perform query expansion such as LCE.

6.5 Evaluation

In this section, we report the results of the empirical evaluation of the parameterized query expansion method (PQE) described in the previous section. We compare the PQE method both to retrieval baselines that do not employ query expansion (Section 6.5.1) and to the latent concept expansion (LCE) method (Section 6.5.2). Then, in Section 6.5.3 we examine the robustness of the PQE retrieval method across queries.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
SD	36.20	25.85	53.82	30.90	31.04	19.37
WSD	36.47	26.09	54.09	31.68*	31.25	20.23*
PQE	39.12* _†	29.06* _†	55.37	33.64* _†	31.98	20.83*

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
SD	35.04	25.62	51.11	27.97	22.97	12.99
WSD	37.05	27.41*	52.21	29.36*	25.31	14.56*
PQE	38.35* _†	29.23* _†	53.89*	31.35* _†	24.84	15.02*

Table 6.3. Comparison of the parameterized query expansion method (PQE) to the non-expanded baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.

All the initial retrieval parameters in the experiments reported in this section are set to the default Indri values, which reflect the best-practice settings. The parameter optimization and the evaluation are done using 3-fold cross-validation. The statistical significance of the differences in the performance of the retrieval methods is determined using a Fisher’s randomized test with 10,000 iterations and $\alpha < 0.05$.

The expansion methods LCE and PQE, unless otherwise noted, use the 25 top retrieved documents for constructing the pseudo-relevant set \mathcal{R} and the 10 highest weighted expansion terms for query expansion. This ensures that all the retrieval methods are relatively efficient, even for large-scale web collections.

We measure the performance using standard retrieval metrics for TREC corpora, as described in Section 4.2. For metrics that use binary relevance judgments, we use precision at the top 20 retrieved documents ($P@20$) and mean average precision across all the queries (MAP). For metrics that use graded relevance judgments, we use normalized discounted cumulative gain and expected reciprocal rank at rank 20 ($NDCG@20$ and $ERR@20$, respectively).

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
SD	11.69	41.78	17.09	43.23	8.80	21.36
WSD	11.68	42.02	17.34	44.06	9.41	22.20
PQE	11.82	44.23[†]	17.27	44.58	8.84	21.94

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
SD	11.76	40.91	15.73	40.97	7.58	17.11
WSD	12.04	42.86*	16.52*	42.47*	8.58	19.58*
PQE	12.35*	44.24*	16.89*	43.06*	8.78	19.01

Table 6.4. Comparison of the parameterized query expansion method (PQE) to the non-expanded baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and † respectively.

6.5.1 Comparison with the Non-Expanded Baselines

In this section, we compare the retrieval performance of the parameterized query expansion method (PQE) to the retrieval performance of two state-of-the-art baselines that do not employ query expansion. The first baseline is the sequential dependence model (SD) first proposed by METZLER and CROFT (2005). The second baseline is the weighted variant of the sequential dependence model (WSD), which is based on the parameterized concept weighting approach (refer to Chapter 5 for the detailed description and the empirical comparison of these two retrieval methods).

Table 6.3 compares the performance of the PQE method with these two baselines, when binary metrics are used for evaluation. Note that in almost all of the cases (except for $P@20$ for the *ClueWeb-B* corpus) the PQE method is superior to both SD and WSD methods. It is never significantly worse than any of the two non-expanded baselines, and in many cases statistically significantly better.

The PQE method demonstrates the largest overall effectiveness gains for the *Robust04* corpus, where it improves the retrieval effectiveness (in terms of *MAP*) by 11% for the $\langle title \rangle$ queries, and by 7% for the $\langle desc \rangle$ queries. In contrast, the weakest

performance of the PQE method is for the *ClueWeb-B* corpus. For the *ClueWeb-B* corpus, PQE does not improve *MAP* by more than 3% for both query types, and these improvements are not statistically significant, when compared to the WSD method (which is the best-performing non-expanded retrieval baseline).

These relative improvements are in line with the nature of these two retrieval corpora. While *Robust04* is a clean and relatively small newswire corpus, *ClueWeb-B* is a large noisy web collection that contains a large number of spam documents (LIN *et al.* 2010). Pseudo-relevance feedback with documents retrieved from the *Robust04* corpus is, thus, much more likely to yield expansion terms that are relevant to the information need expressed by the query and to improve the retrieval performance.

Table 6.5 illustrates this point, by showing side-by-side the expansion terms obtained via pseudo-relevance feedback from the *Robust04* and *ClueWeb-B* corpora for queries “*international art crime*” and “*dangerous vehicles*”. In Table 6.5, the expansion terms from the *Robust04* corpus tend to be more specific and focused on the topic of the query (e.g., *GM* and *Honda* for the query “*dangerous vehicles*”), while the terms retrieved from the *ClueWeb-B* corpus are more vague and general (*project*, *road*, *safety*) and sometimes are either incomprehensible or unrelated to the topic of the query (*rankreason*, *www*).

Comparison using the graded relevance judgments shown in Table 6.4 reveals a similar picture to the comparison in Table 6.3. The improvements are most visible for the *Robust04* corpus, and the performance for the *ClueWeb-B* corpus is never significantly better compared to the non-expanded baselines.

One important thing to note is that the PQE method improves the early precision metrics (*P@20*, *ERR@20*, and *NDCG@20*) to a much lesser degree than the *MAP* metric, which takes into account the entire ranked list. This is due to the fact that the PQE method is a query expansion method and therefore it is likely to improve *recall* by introducing new related terms to the query and retrieving documents that are

<i>international art crime</i>	
<i>Robust04</i>	<i>ClueWeb-B</i>
museum	project
work	rankreason
artist	intern
stolen	www
...	...
<i>dangerous vehicles</i>	
<i>Robust04</i>	<i>ClueWeb-B</i>
car	road
gm	safety
honda	good
battery	ar
...	...

Table 6.5. Comparison of the expansion terms obtained via pseudo-relevance feedback from the *Robust04* and the *ClueWeb-B* collections for queries “*international art crime*” and “*dangerous vehicles*”.

relevant to the information need but contain only few (or none) of the query terms. However, introducing new expansion terms does not necessarily have a significant impact on early precision, since the documents retrieved at the top ranks are likely to contain most of the query terms.

6.5.2 Comparison with the Query Expansion Techniques

Table 6.6 and Table 6.7 demonstrate the experimental comparison of the parameterized query expansion method (PQE) to the latent concept expansion method (LCE), when either binary or graded judgments are used, respectively. In most comparisons, the PQE method is superior to the LCE method. The PQE method is always more effective than the LCE method for the $\langle desc \rangle$ queries, and is superior to the LCE method in 9 out of 12 comparisons for the $\langle title \rangle$ queries.

Similarly to the case of the non-expanded baselines (described in the previous section), PQE has less effect on the retrieval performance (compared to the LCE method) for the *ClueWeb-B* corpus than for the *Robust04* and *Gov2* corpora. While for the

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
LCE	38.37	28.89	54.26	32.59	33.07	20.90
PQE	39.12	29.06	55.37*	33.64*	31.98	20.83

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
LCE	37.29	28.32	51.95	30.34	23.70	14.09
PQE	38.35	29.23*	53.89*	31.35*	24.84	15.02

Table 6.6. Comparison of the parameterized query expansion method (PQE) to the latent concept expansion (LCE) baseline based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method is marked by *.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
LCE	11.84	43.77	16.65	43.26	8.82	21.95
PQE	11.82	44.23	17.27	44.58	8.84	21.94

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
LCE	12.08	42.59	15.88	40.48	8.54	17.90
PQE	12.35	44.24*	16.89*	43.06*	8.78	19.01

Table 6.7. Comparison of the parameterized query expansion method (PQE) to the latent concept expansion (LCE) baseline based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method is marked by *.

		Topics	<i>MAP</i>	Source
(a)	MIX+SOFT-10	351-400	21.25	<i>Table 11</i> (CAO <i>et al.</i> 2008)
	PQE		21.97	
(b)	PRM1	801-850	33.22	<i>Table 2</i> (LV and ZHAI 2010)
	PQE		37.41	

Table 6.8. Comparison of the PQE method with (a) Cao et al., 2008; (b) Lv and Zhai, 2010. Best result per comparison is marked by boldface.

Robust04 and *Gov2* corpora, PQE posits statistically significant gains in 3 out of 4 *MAP* comparisons, it has no significant impact on any retrieval metric, binary or graded, for the *ClueWeb-B* corpus.

This is in line with the results in the previous section. As demonstrated in Table 6.5, the quality of the expansion terms obtained via pseudo-relevance feedback from the *ClueWeb-B* corpus is often low, due to the noisy nature of text contained in the general web documents. Therefore, the parameterized query expansion, which only *re-weights* the expansion terms based on a combination of importance features, is not able to attain significant gains over the non-parameterized latent concept expansion.

In addition to latent concept expansion, we compare the performance of the PQE retrieval method to the performance of two recently proposed query expansion methods that employ some concept weighting and proximity information. The first method was proposed by CAO *et al.* (2008), and uses binary classification to weight expansion terms. The second method was proposed by LV and ZHAI (2010), and leverages term proximities for expansion term weighting. While less general than the approach proposed here, these two methods also focus on concept weighting, and hence we briefly compare their performance to PQE.

For comparison, we use the *MAP* results reported in the papers by CAO *et al.* (2008) and LV and ZHAI (2010), for a subset of topics overlapping with our evaluation. The reported results are for the *<title>* queries only, since these queries are also used

in the papers under consideration. Table 6.8 reports the comparison between the PQE method and these two methods. While we cannot draw statistical significance conclusions, since we have no information on individual query performance, we can see from Table 6.8 that PQE is the best performing method in both comparisons.

In all the cases in Table 6.8 similar query and document processing was applied, and similar baselines were reported. Hence, we can confidently attribute the performance gains to the effectiveness of our method, even when compared to other state-of-the-art query expansion methods that use concept weighting and proximity information.

6.5.3 Robustness

In Table 6.6 and Table 6.7 we have shown that the PQE method significantly improves the overall performance compared to a state-of-the-art latent concept expansion method. In this section, we analyze the *robustness* of the PQE method, compared to the LCE method. Following previous work (METZLER and CROFT 2007a), we define the robustness of the method as the number of queries improved or hurt (and by how much – in terms of *MAP*) as the result of the application of the method. A highly robust expansion technique will significantly improve many queries and only minimally hurt a few.

Figure 6.4 provides an analysis of the robustness of LCE and PQE for the $\langle desc \rangle$ queries. The histograms in Figure 6.4 show, for various ranges of relative decreases or increases in the *MAP* metric, the number of queries that were hurt or improved with respect to a standard bag-of-words baseline, query-likelihood (QL), which is the default retrieval method in Indri. This is in line with the measurement of robustness done by METZLER and CROFT (2007a).

Figure 6.4 unequivocally demonstrates that the PQE method is more robust compared to LCE method. For instance, for the *Robust04* corpus, PQE improves the per-

formance of 73% of the queries w.r.t. QL, compared to 66% of the queries improved by the LCE. Similarly, for the *Gov2* corpus, PQE improves the performance of 73% of the queries w.r.t. QL, compared to 65% of the queries improved by the LCE method. Even for the *ClueWeb-B* corpus, where the performance of the PQE is not better than the LCE performance to a statistically significant degree, PQE improves the performance of 60% of the queries w.r.t. QL, compared to 53% of the queries improved by the LCE.

In addition, the PQE method is much less likely to significantly hurt the performance, compared to the LCE method for the *Robust04* and the *Gov2* corpora. For the *Robust04* corpus, PQE decreases performance by more than 50% for only 5% of the queries, compared to the 7% of the queries hurt by the LCE method. For the *Gov2* corpus, PQE decreases performance by more than 50% for only 6% of the queries, compared to the 10% of the queries hurt by the LCE method. However, for the *ClueWeb-B* corpus, there is no difference in the number of queries significantly hurt by either of the methods.

Finally, it is interesting to examine the relative gains from using the parameterized query expansion compared to the latent concept expansion across all corpora for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Recall that in the previous chapter, we found that while concept weighting is important for all queries, it benefits the longer, more verbose queries to a larger degree due to the fact that they tend to include concepts that have varying importance for expressing the query intent (see Table 5.3 for detailed comparison).

Similarly, in this section we pose a similar hypothesis for the parameterized query expansion. Since the PQE method combines the effects of the parameterized concept weighting of the explicit query concepts and the parameterized weighting of the expansion terms, we hypothesize that the effectiveness gains of the PQE method (compared to the non-parameterized LCE method) will be more pronounced for the verbose $\langle desc \rangle$ queries.

	% queries (50+% gain)	% queries (50+% loss)	% gain
$\langle title \rangle$	4.3	1.3	1.3
$\langle desc \rangle$	15.4	5.1	4.4

Table 6.9. Average effect of the parameterized query expansion (PQE) method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the *MAP* metric).

Table 6.9 examines the difference in effectiveness gains compared to the LCE method (as measured by *MAP*) as a result of applying the PQE method to both $\langle title \rangle$ and $\langle desc \rangle$ queries averaged across the three corpora. Table 6.9 clearly demonstrates that while the parameterized query expansion is beneficial for both types of queries, its effect is much more pronounced for the verbose $\langle desc \rangle$ queries. While it significantly hurts more $\langle desc \rangle$ queries than $\langle title \rangle$ queries (5.1% vs. 1.3%, respectively), it has a significant positive impact (more than 50% effectiveness gain) on more than 15% of $\langle desc \rangle$ queries, compared to slightly more than 4% of the $\langle title \rangle$ queries. In addition, the overall average effectiveness gain as a result of concept weighting is almost three times higher for the $\langle desc \rangle$ queries.

6.6 Summary

In this chapter we introduced the parameterized query expansion using query hypergraphs. First, in Section 6.2, we outlined the theoretical foundations of pseudo-relevance feedback and the state-of-the-art latent concept expansion model (METZLER and CROFT 2007a). Then, in Section 6.3, we showed how the process of parameterized query expansion can be modeled within the query hypergraph framework. In Section 6.4, we specified the parameter optimization in the parameterized query expansion model. In Section 6.5 we empirically evaluated the performance of the parameterized query expansion model.

One important shortcoming of the parameterized query expansion as described in this chapter, is the fact that we only use a single information source, namely the re-

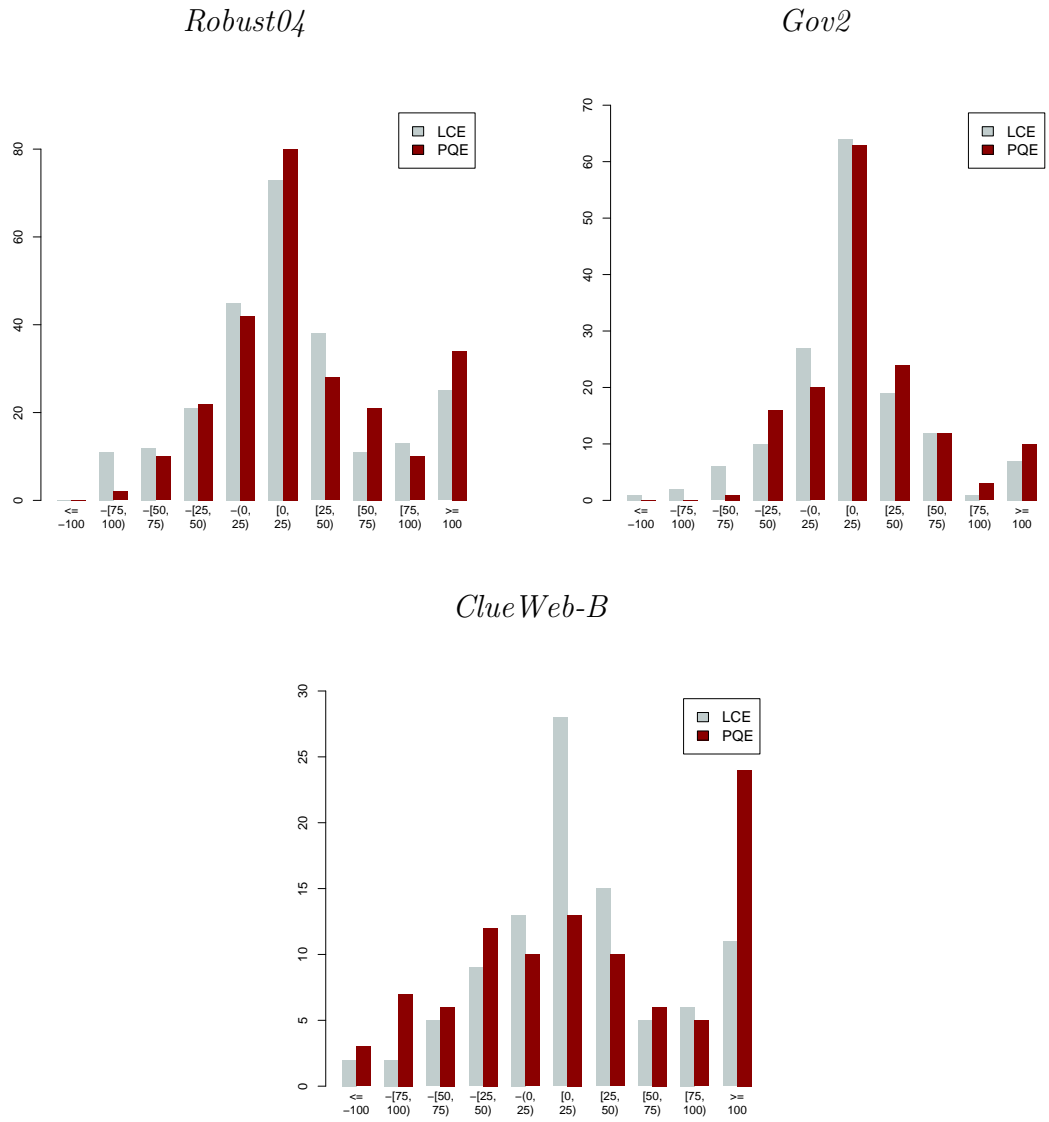


Figure 6.4. Robustness of the LCE and PQE methods for the $\langle desc \rangle$ queries with respect to the QL method.

trieval corpus, for deriving the expansion terms. In the next chapter, we describe how this shortcoming can be addressed by developing a parameterized query expansion approach that leverages and merges evidence from multiple information sources.

CHAPTER 7

PARAMETERIZED QUERY EXPANSION WITH MULTIPLE INFORMATION SOURCES

7.1 Introduction

While pseudo-relevance feedback using the retrieval corpus described in the previous chapter often results in increased retrieval performance, it has a drawback of using only a single information source for performing query expansion. Oftentimes, this approach may lead to a low recall of relevant expansion terms. This is especially true for large-scale web corpora where the quality of the initial set of retrieved documents may be insufficient for generating useful expansion terms.

To illustrate this phenomena, Table 7.1 compares the output of query expansion using multiple sources proposed in this chapter for the keyword query “*ER TV Show*” to the output of the latent concept expansion (LCE) method (METZLER and CROFT 2007a) that uses either the retrieval corpus or the Wikipedia corpus for query expansion (please refer to Section 6.2 for more details on the LCE expansion method). It is clear from Table 7.1 that there are two main advantages of the proposed query expansion with multiple information sources (MSE), compared to the LCE method.

First, the LCE method assumes equal importance among query terms and query phrases by assigning them fixed weights. On the other hand, the proposed MSE method takes a parameterized concept weighting approach and assigns relative importance weights, based on the evidence from multiple importance features, to explicit query terms and phrases. For instance, in the context of the query “*ER TV Show*”, the most important term is “*er*” and the phrase “*er tv*” is more important than the phrase “*tv show*”.

Latent Concept Expansion (*Retrieval Corpus*)
(*Retrieval Corpus*)

Query	Expansion Terms
0.479 er	0.145 tv
0.479 tv	0.112 er
0.479 show	0.055 folge
0.120 er tv	0.054 selbst
0.120 tv show	0.034 show
	...

$AP = 12.29$

Latent Concept Expansion (*Wikipedia*)

Query	Expansion Terms
0.464 er	0.156 tv
0.464 tv	0.074 bisexual
0.464 show	0.066 film
0.116 er tv	0.064 season
0.116 tv show	0.059 series
	...

$AP = 25.68$

Multiple Source Expansion

Query	Expansion Terms
0.297 er	0.085 season
0.168 tv	0.065 episode
0.192 show	0.051 dr
0.051 er tv	0.043 drama
0.012 tv show	0.036 series
	...

$AP = 38.31$

Table 7.1. Comparison of the performance of the latent concept expansion (LCE) with retrieval corpus or Wikipedia to the performance of the query expansion using multiple information sources (MSE) for the query “*ER TV Show*”.

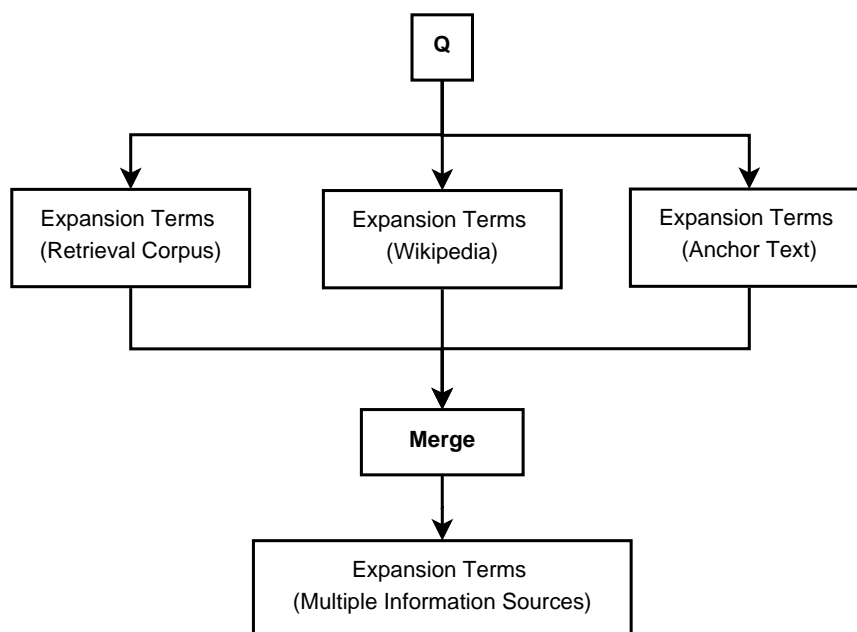


Figure 7.1. Schematic diagram of query expansion with three information sources: retrieval corpus, Wikipedia, and anchor text.

Second, LCE uses a single source for expansion, which can sometimes lead to topic drift. As a case in point, in Table 7.1, LCE with the retrieval corpus expands the query with non-English terms *folge* and *selbst*, and LCE with Wikipedia expands the query with non-helpful terms *bisexual* and *film*. To combat topic drift, the MSE method combines evidence from multiple sources (including, among others, the retrieval corpus and the Wikipedia) to derive a relevant and diverse list of expansion terms.

Note that the MSE expansion method also differs from the PQE method described in the previous chapter. Rather than re-weighting the expansion terms coming from a single source (pseudo-relevance feedback with the retrieval corpus), it assigns weights to multiple information sources, which are used for pseudo-relevance feedback. In this way, MSE may discover diverse, relevant expansion terms that are not returned by the pseudo-relevance feedback using the original corpus.

Due to these advantages, we hypothesize that a parameterized query expansion that uses multiple information sources will yield better results than any of the pre-

viously discussed query expansion methods in isolation. In fact, for the query in Table 7.1, our query expansion improves the retrieval performance by 50% compared to the best performing LCE-based method.

The query expansion method presented in this chapter¹ synthesizes three main research directions. First, it incorporates the highly effective term proximity matching of the sequential dependence model, which was first proposed by Metzler and Croft (METZLER and CROFT 2005). Second, it incorporates the state-of-the-art parameterized concept weighting framework discussed in Chapter 5. Finally, it is inspired by previous work that demonstrates that query expansion using external corpora is highly effective (DIAZ and METZLER 2006; LIN *et al.* 2011; XU *et al.* 2009).

Figure 7.1 shows a schematic diagram of query expansion using multiple information sources. The diagram shows the case of expansion with three information sources, however the same principle may be applied to any number of information sources, without a loss of generality.

First, a query is issued to each of the information sources, and a list of expansion terms is retrieved for each source using pseudo-relevance feedback (see the diagram in Figure 6.1 for a detailed description of the pseudo-relevance feedback process). Then, at the **Merge** stage, the expansion terms from all the sources are combined into a single list. The **Merge** stage takes into account both the expansion source and the term score in the expansion source for determining the final merged score of the expansion term. Finally, the merged list of expansion terms is used for ranking the documents in the collections in response to the user query.

In the remainder of this chapter, we provide details on the process of parameterized query expansion using multiple information sources, as schematically described in Figure 7.1. In Section 7.2 we model the multiple source query expansion using query

¹This chapter is partly based on the work published at the Fifth ACM International Conference on Web Search and Data Mining (BENDERSKY *et al.* 2012).

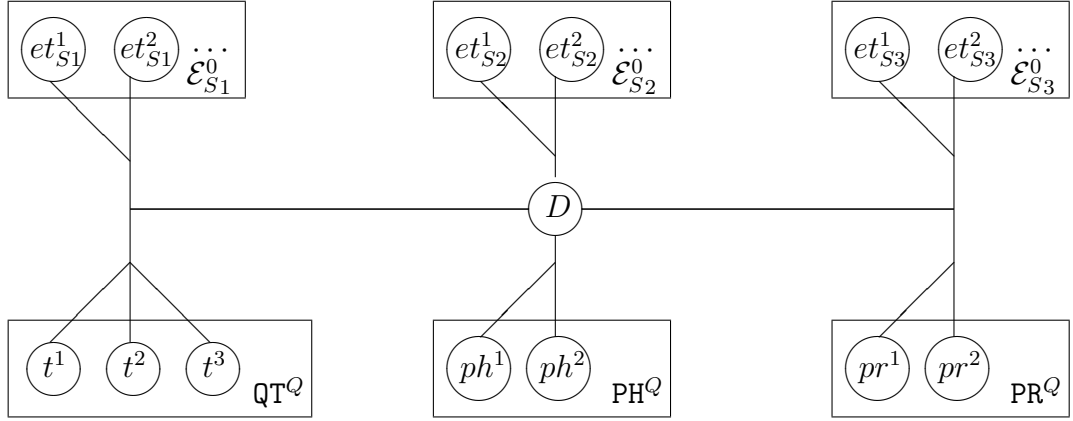
hypergraphs. Then, in Section 7.3, we describe the information sources used for query expansion in this chapter. In Section 7.4, we outline the optimization of the free parameters in the multiple source expansion. In Section 7.5, we report the results of the empirical evaluation of query expansion using multiple information sources. Finally, we conclude this chapter in Section 7.6.

7.2 Multiple Source Expansion with Query Hypergraphs

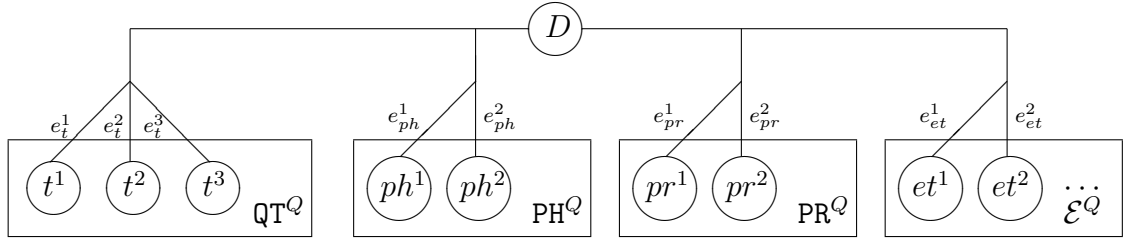
Recall from Section 3.3.3, that the concept importance weight $\lambda(\kappa)$ measures the importance of concept κ for conveying the user intent underlying the query Q . In its simplest form, the concept importance function may be a single collection statistic associated with the concept κ such as inverse document frequency (SPARCK JONES 1988) or the normalized ICF factor (METZLER and CROFT 2007a).

Thus far in this dissertation we have shown that the supervised models of concept weighting that leverage statistics from external information sources (e.g., query logs, Wikipedia, large n-gram repositories, etc.) can significantly improve the retrieval performance. However, these models were used for either weighting the explicit query concepts (as in the weighted sequential dependence model introduced in Chapter 5), or re-weighting the expansion terms that were associated with the query via pseudo-relevance feedback using the retrieval corpus (as in the parameterized query expansion model in Chapter 6).

In contrast, in this section we show that external information sources can also be used, in addition to concept weighting, to select and weight related and helpful terms with which the original query can be expanded. As the example in Table 7.1 demonstrates, such terms can be more relevant and diverse than the expansion terms that are obtained through the standard process of pseudo-relevance feedback on the retrieval corpus, as presented in the previous work by LAVRENKO and CROFT (2003) and METZLER and CROFT (2007a).



(a) Hypergraph H_{FULL}^{MSE} encodes the original query as well as all the expansion terms from a set of sources \mathcal{S} .



(b) Hypergraph H^{MSE} encodes the original query and the highest weighted expansion terms in the \mathcal{E}^Q structure.

Figure 7.2. Two hypergraphs that encode the multiple source expansion model for a three-term query with three information sources.

To this end, we define a set of external information sources \mathcal{S} , which we use as a basis for deriving features for query expansion. To make our approach as widely applicable as possible, we make no assumptions about the internal structure of these sources, and treat them as standard unstructured textual corpora. We defer the precise definition of the external information sources in the set \mathcal{S} used for weighting and expansion to Section 7.3.

In what follows, we explain how to use this set of external sources \mathcal{S} for the expansion of the original query with new related terms. We then show how to construct a hypergraph corresponding to these concepts, and how to rank the documents in the collection accordingly.

Following previous chapters, to assign a weight to an explicit query concept κ we use the parameterized concept weighting approach described in Chapter 5. Recall that in this approach, a parameterized concept weight is expressed as a weighted combination of importance features

$$w_{\text{PCW}}(\kappa) = \sum_{\varphi \in \Phi} \lambda(\varphi) \varphi(\kappa).$$

As shown in Section 5.3, this parameterized concept weighting gives rise to the weighted sequential dependence retrieval model, which assigns a relevance score to document D in response to query Q by a ranking function

$$sc_{\text{WSD}}(Q, D) = \sum_{\sigma \in \{\text{QT,PH,PR}\}} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D),$$

where the set $\{\text{QT,PH,PR}\}$ is a set of structures containing the explicit query concepts (terms, phrases and proximity matches).

A key observation that was made in Chapter 6 is that the proposed ranking function is not limited to the set of explicit query concepts contained in these structures.

Instead, as demonstrated by the parameterized query expansion approach in Chapter 6 the ranking function may include expansion concepts from the retrieval corpus, rather than the search query itself.

In this section, we generalize the definition of expansion concepts to include concepts that are obtained from sources other than the retrieval corpus, as is the standard practice in much of the previous work (LAVRENKO and CROFT 2003; CAO *et al.* 2008; METZLER and CROFT 2005; METZLER and CROFT 2007a). While any combination of terms can serve as an expansion concept, following Chapter 6, in this section we focus on expansion with single terms, mainly for ensuring the efficiency of the expansion concept selection process.

Let \mathcal{S} be a set of external textual sources that can be used for query expansion via pseudo-relevance feedback (see Section 7.3 for an exact definition of these sources). To incorporate expansion terms from the external sources in the set \mathcal{S} , we first obtain a large pool of potential expansion terms associated with each information source $S \in \mathcal{S}$ using pseudo-relevance feedback. To this end, we first rank documents in the source S using the ranking function $sc_{\text{WSD}}(Q, D)$, defined above, which utilizes only explicit query concepts and their corresponding weights.

Then, each term in the pseudo-relevant set of documents \mathcal{R}_S (top ranked documents in source S) is assigned an *expansion score* based on the latent concept expansion weighting described in Equation 6.1.

$$\psi(\kappa, S) = \sum_{D \in \mathcal{R}_S} \exp \left(\gamma_1 sc_{\text{WSD}}(Q, D) + \gamma_2 f(\kappa, D) - \gamma_3 \log \frac{tf_{\kappa, S}}{|S|} \right), \quad (7.1)$$

where γ_i 's are free parameters.

Recall from Section 6.2 that the latent concept expansion score $\psi(\kappa, S)$ is a linear combination of three key components: document relevance (manifested by the document score $sc(Q, D)$), weight of the term in the pseudo-relevant set \mathcal{R}_S (manifested by the matching function $f(\kappa, D)$), and the inverse of the frequency of the term in

the source S ($-\log \frac{tf_{\kappa,S}}{|S|}$), which dampens the scores of very common terms, thereby improving the quality of the expansion terms.

Finally, at most 100 terms with the highest value of $\psi(\kappa, S)$ per source S are added to the initial structure expansion structure \mathcal{E}_S^0 , which contains the initial pool of expansion terms associated with source S . The large number of expansion terms in the initial pool \mathcal{E}_S^0 ensures that it is large enough for selecting diverse expansion terms at the second stage. Note that it is guaranteed that the total number of expansion terms in all sources is bounded by $100|\mathcal{S}|$.

Once the initial expansion term structures \mathcal{E}_S^0 are obtained, we assign a weight to each of the unique expansion terms in these structures

$$\kappa \in \bigcup_{S \in \mathcal{S}} \mathcal{E}_S^0,$$

using the weighted combination of expansion scores

$$w_{\text{MSE}}(\kappa) = \sum_{S \in \mathcal{S}} \lambda(S) I(\kappa, S), \quad (7.2)$$

where $I(\kappa, S)$ is an indicator function defined as

$$I(\kappa, S) = \begin{cases} \psi(\kappa, S) & \text{if } \kappa \in \mathcal{E}_S^0 \\ 0 & \text{else} \end{cases}$$

According to Equation 7.2, the weight $w_{\text{MSE}}(\kappa)$ is expressed by a weighted combination of *expansion scores*, which are defined over a set of sources \mathcal{S} . Each expansion score $\psi(\kappa, S)$ is associated with an expansion term κ and is computed over a source $S \in \mathcal{S}$. To handle missing terms, if κ is not one of the top 100 terms selected from the source S , we set $I(\kappa, S) = 0$.

To ensure efficient query expansion, we retain only the top 10 terms from the set of expansion terms $\bigcup_{S \in \mathcal{S}} \mathcal{E}_S^0$, based on Equation 7.2. We refer to this small set of expansion terms as \mathcal{E}^Q .

The hypergraphs H_{FULL}^{MSE} and H^{MSE} depicted in Figure 7.2 graphically represent this expansion process. The full hypergraph H_{FULL}^{MSE} (Figure 7.2(a)) includes all the expansion terms from the set of all the sources $\bigcup_{S \in \mathcal{S}} \mathcal{E}_S^0$. For efficiency reasons, only a small set of highest weighted expansion terms encoded in the structure \mathcal{E}^Q in the hypergraph H^{MSE} (Figure 7.2(b)) is used for ranking the documents in the collection.

Following these definitions of the explicit concept weights and the expansion term weights, the ranking function for the multiple source expansion (MSE) approach becomes:

$$\begin{aligned}
sc_{MSE}(Q, D) &\triangleq sc_{WSD}(Q, D) + \sum_{\kappa \in \mathcal{E}^Q} w_{MSE}(\kappa) f(\kappa, D) = \\
&= \sum_{\sigma \in \{QT, PH, PR\}} \sum_{\varphi \in \Phi} \lambda(\varphi, \sigma) \sum_{\kappa \in \sigma} \varphi(\kappa) f(\kappa, D) + \\
&+ \sum_{S \in \mathcal{S}} \lambda(S) \sum_{\kappa \in \mathcal{E}^Q} I(\kappa, S) f(\kappa, D). \tag{7.3}
\end{aligned}$$

To complete the derivation of this ranking function, in Section 7.3 we describe the set of external sources \mathcal{S} used for query expansion. Then, in Section 7.4 we describe the pipeline optimization process for optimizing the weights Λ in Equation 7.3.

7.3 Information Sources

In this section, we provide a detailed description of the set of external information sources \mathcal{S} used for query expansion. As described in Section 7.2, we make no assumptions about the internal structure of these sources, and treat them as unstructured textual corpora. We use these external information sources to perform pseudo-relevance feedback for computing the expansion scores associated with the expansion terms in the structure \mathcal{E}^Q .

<i>Information Source</i>	<i>Unit of Retrieval</i>
Retrieval Corpus	Single document
Wikipedia Corpus	Single article
<i>ClueWeb-B</i> Anchor Text	Single line of anchor text (as defined by the < <i>a</i> > HTML tag)
<i>ClueWeb-B</i> Heading Text	Single line of heading text (as defined by the < <i>h*</i> > HTML tags)

Table 7.2. External information sources used in the multiple source expansion (MSE) method.

Retrieval Corpus chemical, weapon, toxic, convention, substance, gas, destruction, product, plant, mirzayanov, ...
Wikipedia Corpus chemical, agent, gas, weapon, warfare, war, poison, mustard, disseminate, nerve, ...
<i>ClueWeb-B</i> Anchor Text toxic, chemical, cigarette, tobacco, terrorist, tts, weapon, leach, terror, wwf, ...
<i>ClueWeb-B</i> Heading Text toxic, chemical, weapon, terrorist, terror, assess, biology, behavior, incinerate, emission, ...
MSE weapon, agent, gas, russia, convention, mustard, warfare, substance, destruction, product, ...

Table 7.3. Comparison between the lists of expansion terms derived from the individual external information sources for the query “*toxic chemical weapon*” and the combined list produced by the MSE method.

It is theoretically possible to use the same information sources for deriving both the set of importance features described in Section 3.4.2 and the expansion scores. In practice, however, a single external source is commonly better suited for only one of these tasks. For instance, the Google N-grams source (a large collection of web n-gram counts) is useful for concept weighting, but not for query expansion. On the other hand, an entire external document collection such as Wikipedia is more suitable for query expansion.

Accordingly, in Table 7.2 we provide a list of external information sources used for query expansion along with a brief description of their utilization. Table 7.2 defines a unit of retrieval, which is used for pseudo-relevance feedback from the source. As external sources for query expansion, we use, in addition to the retrieval corpus, the heading text and the anchor text extracted from the TREC collection *ClueWeb-B*, a large, publicly available web collection used as a dataset in our experiments (see Chapter 4 for more details about this collection), as well as an English Wikipedia corpus.

As an example of the role that the external sources may play in query formulation, Table 7.3 demonstrates the expansion terms derived from the external information sources for the query *“toxic chemical weapon”*. The MSE column in Table 7.3 is the output of the process of expansion with multiple information sources described in Section 7.2. The MSE column includes expansion terms which are more relevant and address more of the query aspects than those produced by any individual source.

For instance, MSE expansion includes the terms *russia*, *agent*, *mustard* and *warfare*, which do not appear in the top terms obtained via pseudo-relevance feedback on the retrieval corpus. As a result, in this case, the MSE approach improves the retrieval effectiveness by 33% over a method that uses latent concept expansion with the retrieval corpus, and by 14% over a method that uses latent concept expansion with Wikipedia.

MSEOptimization(Λ^0)

- 1: $\Lambda_Q^0 \leftarrow \Lambda_{\{\text{QT,PH,PR}\}}^0$
- 2: $\Lambda_S^0 \leftarrow \{\lambda_S^0 : S \in \mathcal{S}\}$
- 3: $\langle \mathcal{M}, \Lambda_Q \rangle \leftarrow \text{CoordinateAscent}(\emptyset, \Lambda_Q^0)$
- 4: $\langle \mathcal{M}, \Lambda_S \rangle \leftarrow \text{CoordinateAscent}(\Lambda_Q, \Lambda_S^0)$
- 5: **return** $\langle \mathcal{M}, \Lambda_Q \cup \Lambda_S \rangle$

Figure 7.3. Pipeline optimization of the multiple source expansion method.

7.4 Parameter Optimization

Similarly to the case of the parameterized query expansion with the retrieval corpus (discussed in Section 6.4), the optimization of the multiple source expansion is performed in several stages. Therefore, for the optimization of the free parameters Λ in Equation 7.3 we employ an optimization procedure, which is based on the pipeline optimization discussed in Section 3.5.3. This procedure is outlined in Figure 7.3.

First, we denote the initial parameterization of the explicit concepts (concepts in the QT, PH, and PR structures) Λ_Q^0 , and the initial parameterization of the expansion sources Λ_S^0 . Then, we optimize the weights of the explicit query concepts alone, using the coordinate ascent algorithm (see Figure 3.3). This process yields an *optimized* parameterization Λ_Q , which is then used to obtain a list of expansion terms from each of the information sources in \mathcal{S} using pseudo-relevance feedback.

As the initial parameterization of the expansion sources, we set

$$\lambda(\text{Retrieval Corpus}) = 1,$$

and the rest of the parameters to 0. This ensures that the starting point of our optimization is exactly the latent concept expansion approach with optimized explicit concept weights. In this manner, if the additional sources are deemed not to be helpful for expansion, they will not contribute any expansion terms to the expansion structure \mathcal{E}^Q used in the ranking function. Once all the expansion terms are collected, the set

of free parameters associated with the expansion sources in the set \mathcal{S} is optimized using the coordinate ascent algorithm.

It is computationally infeasible to use *all* the expansion terms from *all* the expansion sources at each iteration of the coordinate ascent algorithm. To alleviate this problem to some degree, and to make the optimization process more efficient, at each iteration of the coordinate ascent algorithm, we include in the expanded queries at most 10 expansion terms with the highest weight (as determined by the parameterization $\Lambda_{\mathcal{S}}^i$ at the i -th iteration of the coordinate ascent algorithm), which is referred to as the \mathcal{E}^Q structure in the H^{MSE} representation in Figure 7.2(b).

The optimization phase concludes after the second round of the coordinate ascent algorithm is completed. At this point, the entire set of weights Λ is optimized in terms of the target retrieval metric \mathcal{M} .

In this way, we ensure that the parameters Λ in the MSE ranking function (Equation 6.3) are optimized to deliver both the best selection of the expansion terms and the most effective retrieval performance of the expanded queries. As our experimental results demonstrate, this leads to a significant improvement over the state-of-the-art non-parameterized retrieval methods that perform query expansion such as LCE, as well as the parameterized query expansion method (PQE) described in Chapter 6 that uses only a single expansion source.

7.5 Evaluation

In this section, we report the results of the empirical evaluation of query expansion using multiple information sources (MSE) described in the previous section. We compare the MSE method both to retrieval baselines that do not employ query expansion (Section 7.5.1) and to the latent concept expansion (LCE) and the parameterized query expansion (PQE) methods (Section 7.5.2). Then, in Section 6.5.3 we examine the robustness of the MSE retrieval method across queries.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
SD	36.20	25.85	53.82	30.90	31.04	19.37
WSD	36.47	26.09	54.09	31.68*	31.25	20.23*
MSE	38.80* [†]	30.49* [†]	56.59* [†]	34.35* [†]	39.17* [†]	23.96* [†]

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
SD	35.04	25.62	51.11	27.97	22.97	12.99
WSD	37.05	27.41*	52.21	29.36*	25.31	14.56*
MSE	38.90* [†]	30.68* [†]	54.46* [†]	31.10* [†]	26.87* [†]	15.23* [†]

Table 7.4. Comparison of the parameterized query expansion methods to the non-expanded baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.

All the initial retrieval parameters in the experiments reported in this section are set to the default Indri values, which reflect the best-practice settings. The parameter optimization and the evaluation are done using 3-fold cross-validation. The statistical significance of the differences in the performance of the retrieval methods is determined using a Fisher’s randomized test with 10,000 iterations and $\alpha < 0.05$.

The expansion methods LCE PQE, and MSE, unless otherwise noted, use the 25 top retrieved documents for constructing the pseudo-relevant set \mathcal{R} and the 10 highest weighted expansion terms for query expansion. This ensures that all the retrieval methods are relatively efficient, even for large-scale web collections.

We measure the performance using standard retrieval metrics for TREC corpora, as described in Section 4.2. For metrics that use binary relevance judgments, we use precision at the top 20 retrieved documents ($P@20$) and mean average precision across all the queries (MAP). For metrics that use graded relevance judgments, we use normalized discounted cumulative gain and expected reciprocal rank at rank 20 ($NDCG@20$ and $ERR@20$, respectively).

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
SD	11.69	41.78	17.09	43.23	8.80	21.36
WSD	11.68	42.02	17.34	44.06	9.41	22.20
MSE	11.86	44.13* _†	17.33	44.91	9.94	25.76* _†

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
SD	11.76	40.91	15.73	40.97	7.58	17.11
WSD	12.04	42.86*	16.52*	42.47*	8.58	19.58*
MSE	12.62* _†	44.86* _†	16.67	43.13*	8.59	20.32*

Table 7.5. Comparison of the parameterized query expansion methods to the non-expanded baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the SD method and the WSD method are marked by * and †, respectively.

7.5.1 Comparison with the Non-Expanded Baselines

In this section, we compare the retrieval effectiveness of query expansion with multiple information sources MSE, which performs both concept weighting and query expansion to the performance of the methods that perform query weighting alone. The first baseline is the sequential dependence model (SD) first proposed by METZLER and CROFT (2005). The second baseline is the weighted variant of the sequential dependence model (WSD), which is based on the parameterized concept weighting approach (refer to Chapter 5 for the detailed description and the empirical comparison of these two retrieval methods).

Table 7.4 and Table 7.5 compare the performance of the above baselines (SD and WSD) and query expansion with multiple information sources MSE. Both tables unequivocally demonstrate the effectiveness of query expansion with multiple sources. In all but one comparisons, MSE is more effective than the baselines that do not perform query expansion, and in many of the cases (especially in the case of the MAP metric) its improvements are statistically significant. These improvements are consistent across retrieval metrics, corpora and query types.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
LCE	38.37	28.89	54.26	32.59	33.07	20.90
LCE-WP	38.94	28.93	53.75	31.90	39.22* _‡	23.48 _‡
PQE	39.12	29.06	55.37* _‡	33.64* _‡	31.98 _‡	20.83 _‡
MSE	38.80	30.49* _{‡‡}	56.59* _{‡‡}	34.35* _{‡‡}	39.17* _‡	23.96* _‡

$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
LCE	37.29	28.32	51.95	30.34	23.70	14.09
LCE-WP	38.33	29.08	51.31	28.70	26.56	14.52
PQE	38.35	29.23*	53.89* _‡	31.35* _‡	24.84 _‡	15.02
MSE	38.90*	30.68* _{‡‡}	54.46* _‡	31.10 _‡	26.87* _{‡‡}	15.23* _{‡‡}

Table 7.6. Comparison of the parameterized query expansion methods to the query expansion baselines based on the binary relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method, the LCE-WP method and the PQE methods are marked by *, †, and ‡ respectively.

Recall that in the Section 6.5.1, we have shown that parameterized query expansion using the retrieval corpus fails to improve retrieval effectiveness over the SD and the WSD methods for the *ClueWeb-B* corpus (refer to Table 6.3 and Table 6.4 for detailed comparisons). In contrast, MSE method is always more effective than the two non-expanded baselines both for binary and graded metrics. The effectiveness improvements for the $\langle title \rangle$ queries (as measured by the *MAP* metric) are statistically significant. This observation showcases the importance of using external information sources in the pseudo-relevance feedback process, when the retrieval corpus is a large-scale noisy web collection.

7.5.2 Comparison with the Query Expansion Techniques

After comparing the effectiveness of the MSE method against methods that do not perform query expansion, in this section we focus on comparing its performance to that of current state-of-the-art query expansion methods.

$\langle title \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
LCE	11.84	43.77	16.65	43.26	8.82	21.95
LCE-WP	12.22	44.63	17.47	43.98	9.69	25.43
PQE	11.82	44.23	17.27	44.58	8.84	21.94
MSE	11.86	44.13	17.33	44.91	9.94	25.76* _‡
$\langle desc \rangle$	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
LCE	12.08	42.59	15.88	40.48	8.54	17.90
LCE-WP	12.55	44.38	16.02	41.45	8.67	19.90
PQE	12.35	44.24*	16.89*	43.06*	8.78	19.01
MSE	12.62*	44.86*	16.67	43.13*	8.59	20.32*

Table 7.7. Comparison of the parameterized query expansion methods to the query expansion baselines based on the graded relevance metrics for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Best result in the column is bolded. Statistically significant differences with the LCE method, the LCE-WP method and the PQE methods are marked by *, †, and ‡ respectively.

First, we make use of the latent concept expansion method, which was shown to be a state-of-the query expansion method that uses a single collection (METZLER and CROFT 2007a; LANG *et al.* 2010). See Section 6.2 for a detailed description of the LCE method.

As baselines, we implement two variants of latent concept expansion. The first baseline is denoted LCE. It is the standard version of latent concept expansion, which performs the pseudo-relevance feedback on the retrieval corpus.

The second baseline is denoted LCE-WP. LCE-WP performs the pseudo-relevance feedback on Wikipedia, rather than the retrieval corpus. LCE-WP is based on some recent work that shows that query expansion using Wikipedia corpus can be beneficial, especially for short ambiguous queries over large web collections (LI *et al.* 2007; XU *et al.* 2009).

In addition to the LCE-based baselines, we use the parameterized query expansion method described in Chapter 6 as a baseline. Recall that the PQE method, combines explicit concept weighting and expansion term weighting in a unified framework that

uses external information sources. The main difference between the PQE and the MSE methods, is that the former uses the external sources solely for weighting purposes, while the latter uses them also for expansion term selection.

Table 7.6 and Table 7.7 compare the effectiveness of the three baselines described above (LCE, LCE-WP and PQE) to the proposed MSE method using binary and relevance metric, respectively. This comparison highlights the different positive aspects of MSE method.

The main observation from Table 7.6 and Table 7.7 is that MSE is in many cases more effective than any of the three baselines (e.g., it is the most effective method in terms of *MAP* in all but one comparisons). In contrast to the baselines, the performance of MSE is stable across corpora and query types. In comparison, the performance of the baselines is not as consistent. For instance, LCE-WP is more effective than LCE for the *Robust04* and *ClueWeb-B* corpora, but less effective for the *Gov2* corpus. Similarly, PQE outperforms LCE-based baselines for *Robust04* and *Gov2* corpora, but is not as effective for the *ClueWeb-B* corpus.

In addition, Table 7.6 and Table 7.7 clearly demonstrate the importance of using external information sources for both concept weighting and expansion term selection. Compared to PQE, which uses the external sources of information solely for weighting purposes, MSE achieves significantly better performance on all metrics. This is especially evident in the case of the *ClueWeb-B* corpus, for which expansion using the retrieval corpus attains only marginal gains. For the *ClueWeb-B* corpus, PQE achieves merely a 3% gain over the WSD baseline for *<title>* queries, while MSE achieves over 18% gain. It is clear that in this case, using multiple sources for selecting the expansion terms, in addition to concept weighting, is highly beneficial.

Finally, Table 7.6 and Table 7.7 shows that the synergy of concept weighting and expansion term selection using external sources as performed by the MSE is superior to the ad-hoc approach that simply uses an external corpus (e.g., Wikipedia) for query

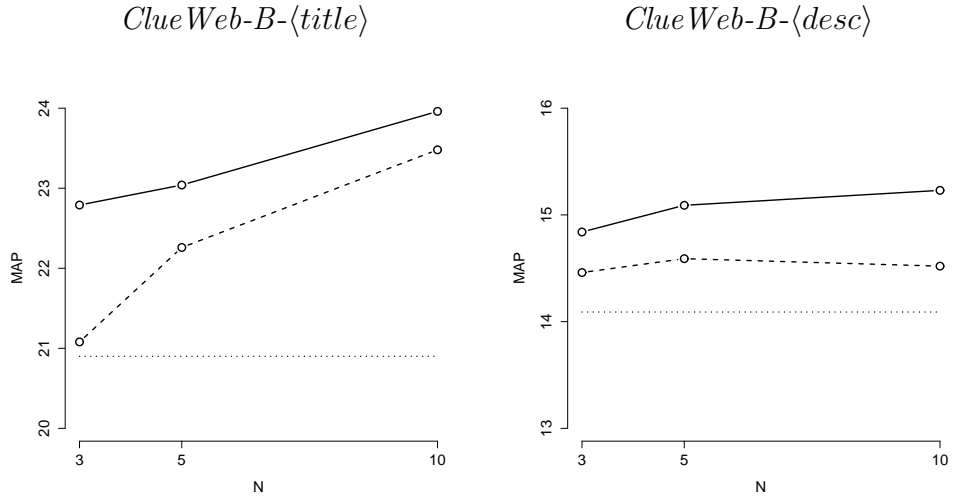


Figure 7.4. Varying the number of expansion terms (*ClueWeb-B* corpus). Dotted line indicates the performance of LCE[10]. Dashed and solid lines represent the performance of LCE-WP[N] and MSF[N], respectively.

expansion. MSE is more stable than LCE-WP across all collections, and is more effective even for the *ClueWeb-B* corpus, where expansion with Wikipedia was shown to be a highly effective strategy (BENDERSKY *et al.* 2011; MCCREADIE *et al.* 2010).

7.5.3 Number of Expansion Terms

Massive query expansion with tens or even hundreds of terms, as is often done in TREC evaluation (CAO *et al.* 2008; DIAZ and METZLER 2006) is not suitable for the scenario of web search, where the size of the retrieval corpus is large, and users expect low query latencies. Accordingly, in this section we explore the effect of query expansion with very few expansion terms, to demonstrate the scalability of the MSE method for web corpora.

In Figure 7.4 we plot the effectiveness (in terms of *MAP*) of query formulation methods that have the best performance for the *ClueWeb-B* corpus – LCE-WP and MSE – when using the 3, 5 and 10 highest weighted expansion terms. For comparison, we also plot the effectiveness of a standard query expansion method, LCE with 10 terms.

$\langle title \rangle$	α -nDCG@20	S-Recall@20	MAP-IA
WSD	22.36	46.04	9.20
PQE	21.07	42.47	9.39
LCE-WP	24.51	46.91	11.00
MSE	25.85 ^{*_‡}	48.94 _‡	11.25 [*] _‡

Table 7.8. Result diversification performance (*ClueWeb-B*). Statistically significant difference of MSE over the baselines are marked using *, †, and ‡, for WSD, PQE and LCE-WP baselines, respectively. Best result per column is marked by boldface.

First, Figure 7.4 clearly demonstrates the superiority of both LCE-WP and MSE compared to LCE, even with fewer expansion terms. We can also see from Figure 7.4 that the superiority of the proposed MSE method over the LCE-WP method, which uses Wikipedia for query expansion, is not limited to the scenario in Table 7.6 and Table 7.7, where 10 expansion terms are used. The effectiveness gains of MSE over LCE-WP are consistent with minimal query expansion (3 or 5 additional terms) as well. For instance, when only 3 terms are used for query expansion, MSE achieves around 8% and 3% improvement over LCE-WP for $\langle title \rangle$ and $\langle desc \rangle$ queries, respectively.

Overall, the results in Figure 7.4 showcase the ability of the MSE method to produce both effective and compact queries, which could potentially scale to real world web search scenarios.

7.5.4 Impact on result diversification

Recently, result diversification in web search has become an active research topic (AGRAWAL *et al.* 2009; CLARKE *et al.* 2008; CLARKE *et al.* 2010; SANTOS *et al.* 2010; SANTOS *et al.* 2011). Since web search queries are often underspecified and/or ambiguous, diversifying the search results may assist users with varying intents in finding relevant information in a single ranked list returned by the search engine. Due to the research interest in this problem, result diversification was chosen as a search task during the 2009 and 2010 TREC Web Tracks (CLARKE *et al.* 2010).

Effective result diversification is often achieved by *inter-query* approaches. These approaches combine results from queries that are found to be related to the original user query (e.g., through access to the query suggestions proposed by commercial search engines (SANTOS *et al.* 2010; SANTOS *et al.* 2011)). However, even in the inter-query approaches, the retrieval effectiveness and diversity performance of each single query is important for obtaining the optimal diversification results (SANTOS *et al.* 2011).

Therefore, in this section we examine *intra-query* result diversification, i.e., the diversity performance that can be achieved by using the original user query alone. To this end, we compare the performance of the three best-performing baselines from Table 6.3 and Table 6.6 (WSD, PQE and LCE-WP) to that of the MSE method in terms of three standard diversity metrics. These diversity metrics include metrics that examine the diversity at the top ranks (α -NDCG and subtopic recall at rank 20) (CLARKE *et al.* 2008; CLARKE *et al.* 2010), as well as a metric that measures the diversity of the entire ranked list (intent-aware mean average precision) (AGRAWAL *et al.* 2009).

Table 7.8 demonstrates the comparison of the result diversification performance of the different methods on the *<title>* queries for the *ClueWeb-B* collection². Overall, MSE achieves the best diversity performance, especially for the diversity at the top ranks, where it achieves over 6% improvement over LCE-WP, the best-performing baseline.

In the context of search result diversification, it is interesting to note that previous work suggested that query expansion with the retrieval corpus may reduce diversity at top ranks (CLARKE *et al.* 2008). The comparison between the WSD and the PQE baselines in Table 7.8 is in line with this finding. In contrast to the expansion with the retrieval corpus alone, the proposed MSE method helps to improve the diversity

²We do not include the *<desc>* queries in our diversification performance analysis, since these are verbose and non-ambiguous queries that fully specify the user intent.

	% queries (50+% gain)	% queries (50+% loss)	% gain
$\langle title \rangle$	13.7	4.4	8.5
$\langle desc \rangle$	19.6	8.1	6.3

Table 7.9. Average effect of the parameterized query expansion (MSE) method on the $\langle title \rangle$ and the $\langle desc \rangle$ queries across all the TREC corpora (as measured by the *MAP* metric).

of the search results, since it combines expansion terms from different information sources.

7.5.5 Robustness

In this section, we analyze the *robustness* of the MSE method, compared to the LCE method. Similarly to Section 6.5.3, we define the robustness of the method as the number of queries improved or hurt (and by how much – in terms of *MAP*) as the result of the application of the method. A highly robust expansion technique will significantly improve many queries and only minimally hurt a few.

Figure 7.5 provides an analysis of the robustness of LCE and MSE for the $\langle desc \rangle$ queries. The histograms in Figure 7.5 show, for various ranges of relative decreases or increases in the *MAP* metric, the number of queries that were hurt or improved with respect to a standard bag-of-words baseline, query-likelihood (QL), which is the default retrieval method in Indri. This is in line with the measurement of robustness done by METZLER and CROFT (2007a).

Figure 7.5 unequivocally demonstrates that the MSE method is more robust compared to LCE method. For instance, for the *Robust04* corpus, MSE improves the performance of 72% of the queries w.r.t. QL, compared to 66% of the queries improved by the LCE. Similarly, for the *Gov2* corpus, MSE improves the performance of 73% of the queries w.r.t. QL, compared to 65% of the queries improved by the LCE method. For the *ClueWeb-B* corpus these improvements are 58% and 53%, respectively.

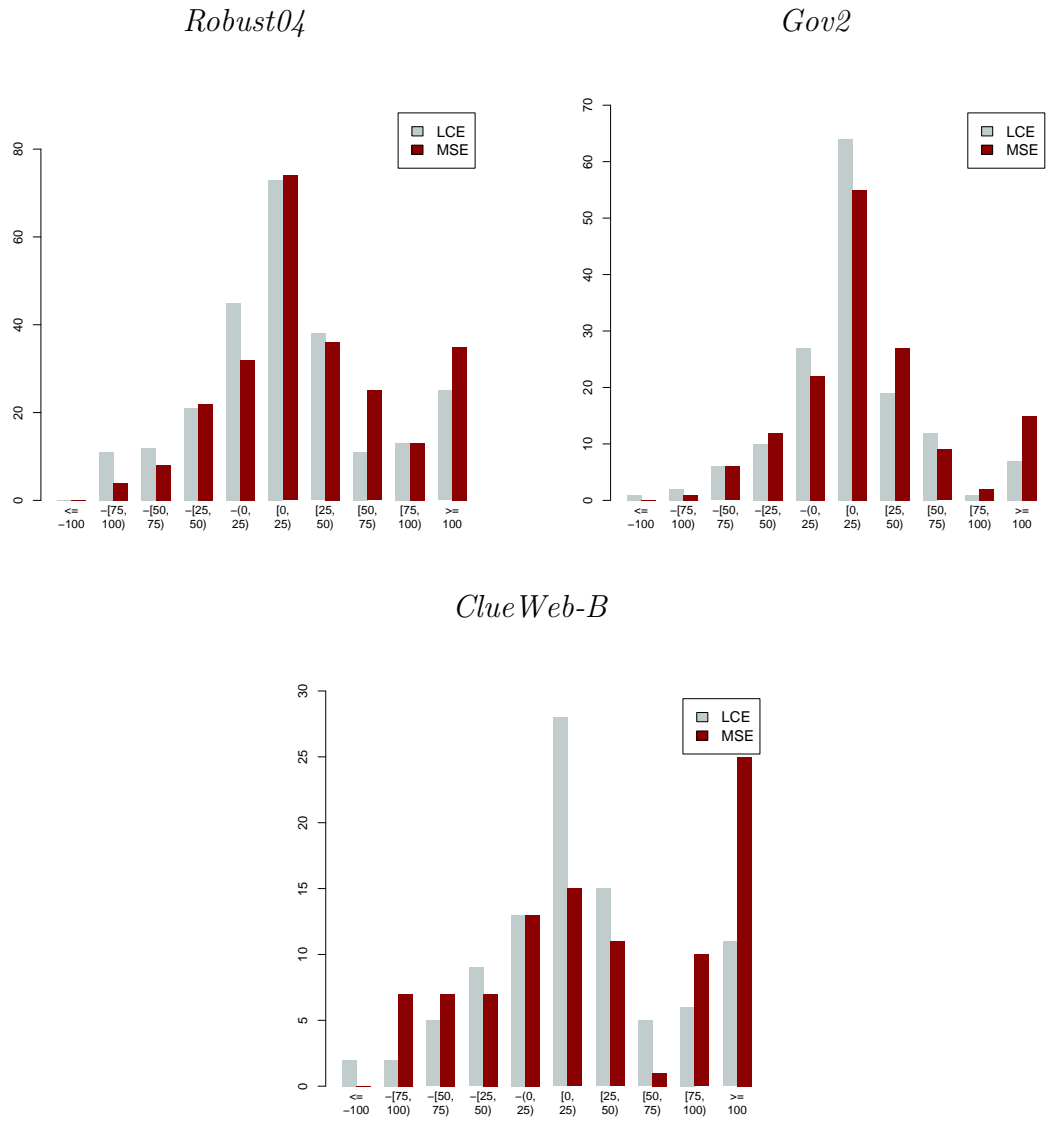


Figure 7.5. Robustness of the LCE and MSE methods for the $\langle desc \rangle$ queries with respect to the QL method.

In addition, the MSE method is much less likely to significantly hurt the performance, compared to the LCE method for the *Robust04* and the *Gov2* corpora. For the *Robust04* corpus, MSE decreases performance by more than 50% for only 4% of the queries, compared to the 7% of the queries hurt by the LCE method. For the *Gov2* corpus, MSE decreases performance by more than 50% for only 5% of the queries, compared to the 10% of the queries hurt by the LCE method.

For the *ClueWeb-B* corpus, MSE decreases performance by more than 50% for slightly more queries than LCE: 17% of the queries, compared to 16%. However, this difference is more than offset by the percentage of queries for which the MSE method improves performance by more than 50%: 33% of the queries, compared to 20% of the queries improved to the same degree by the LCE method.

Finally, it is interesting to examine the relative gains from using the parameterized query expansion compared to the latent concept expansion across all corpora for the $\langle title \rangle$ and the $\langle desc \rangle$ queries. Recall that in the previous chapters, we found that while parameterized concept weighting and expansion is important for all queries, it benefits the longer, more verbose queries to a larger degree due to the fact that they tend to include concepts that have varying importance for expressing the query intent (see Table 5.3 and Table 6.9 for detailed comparisons).

Table 7.9 examines the difference in effectiveness gains compared to the LCE method (as measured by *MAP*) as a result of applying the MSE method to both $\langle title \rangle$ and $\langle desc \rangle$ queries averaged across the three corpora. Table 7.9 clearly demonstrates that query expansion with multiple information sources is beneficial for both types of queries. While, in general, it significantly improves more $\langle desc \rangle$ queries, the overall gain in retrieval performance is comparable among the $\langle title \rangle$ and $\langle desc \rangle$ query types.

7.6 Summary

In this chapter we described the process of parameterized query expansion using multiple information sources. In Section 7.2 we modeled the multiple source parameterized query expansion using query hypergraphs. Then, in Section 7.3, we described the information sources used for query expansion in this chapter. In Section 7.4, we outlined the optimization of the free parameters in the multiple source query expansion. In Section 7.5, we reported the results of the empirical evaluation of query expansion using multiple information sources.

This chapter concludes the exploration of query expansion with query hypergraphs that we began in Chapter 6. This exploration led to two important findings. First, in Chapter 6 we found that the parameterized query expansion approach is significantly more effective than the current state-of-the-art query expansion techniques such as latent concept expansion (METZLER and CROFT 2007a). Second, in this chapter, we found that the effectiveness of the parameterized query expansion with query hypergraphs can be further improved by incorporating evidence from external information sources such as Wikipedia or anchor text.

In the next chapter, we return to examining the retrieval performance of query hypergraphs that do not utilize any query expansion. In particular, in the next chapter we focus on modeling parameterized concept dependencies using query hypergraphs. The parameterized concept dependencies can model dependencies between arbitrary concepts in the query, and assign weights to these dependencies. This is an important advance compared to the current retrieval models that can only model dependencies between single query terms. As we show in the next chapter, both parameterized concept weighting and parameterized concept dependencies can be integrated into a unified retrieval framework based on the query hypergraph representation.

CHAPTER 8

PARAMETERIZED CONCEPT DEPENDENCIES

8.1 Introduction

In the previous chapters, we focused on the incorporation of the parameterized concept weighting into the retrieval function. The weighting was applied to arbitrary concepts, or term dependencies, rather than single query terms. Some additional recent examples of retrieval models that incorporate term dependencies include, among others, Markov random fields (METZLER and CROFT 2005), linear discriminant model (GAO *et al.* 2005), dependence language model (GAO *et al.* 2004), quasi-synchronous dependence model (PARK *et al.* 2011), and positional language model (LV and ZHAI 2009).

However, both the previous chapters of this dissertation, and most of the previous work make the assumption that there are no further dependencies between the concepts in the query, and treats them independently. This approach ultimately leads to *bag-of-concepts* retrieval models.

In this chapter¹, we demonstrate that the query hypergraphs can remedy this shortcoming of the current retrieval models that incorporate term dependencies. Based on this observation, we propose several novel retrieval methods that take a further step toward a more accurate modeling of the dependencies between the query terms. Rather than modeling the dependencies between the *individual query terms*, our retrieval methods model dependencies between *arbitrary concepts* in the query.

¹This chapter is partly based on the work to appear at the 35th Annual ACM SIGIR Conference (BENDERSKY and CROFT 2012).

...linking **law enforcement** duties to the definition of “**law enforcement officer**” for retirement purposes....must be handled within the context of...FEPCA and **law enforcement** retirement law and regulations....Adding a discussion of these issues would add unnecessarily to the complexity...of **information** already **provided**...definitions of “**law enforcement officer**” in these regulations should provide guidance...

(a)

...Simi Valley, West Covina and Los Angeles police departments were among the first **law enforcement** agencies to receive money through the forfeiture program....a narcotics-sniffing **dog** in a Simi Valley police investigation...led to the largest seizure of cocaine ever by authorities from Ventura County...**dog**'s efforts are expected to yield a substantial amount of money...for the 21-officer department...

(b)

Figure 8.1. Excerpts from (a) the top document retrieved by the sequential dependence model, and (b) the top document retrieved using a query hypergraph in response to the query: “Provide information on the use of dogs worldwide for law enforcement purposes”. Non-stopword query terms are marked in boldface.

As described in Section 3.1, we broadly define a query concept as a syntactic expression that models a dependency between a subset of query terms. Query concepts may model a variety of linguistic phenomena, including n-grams, term proximities, noun phrases, and named entities. Therefore, a dependency between query concepts represents a *dependency between term dependencies*, i.e., a *higher-order term dependency*. In the remainder of this chapter, we shall use the definitions “*higher-order term dependency*” and “*concept dependency*” interchangeably.

To the best of our knowledge, there is little prior work on modeling this type of higher-order term dependencies for information retrieval. Most retrieval models limit their attention to either pairwise term dependencies (CUMMINS and O’RIORDAN 2009; LV and ZHAI 2009) or, at most, dependencies between multiple terms (BENDERSKY and CROFT 2008; METZLER and CROFT 2005). In contrast, the query hypergraphs can model dependencies between arbitrary concepts, e.g., a dependency between a phrase and a term, via the inclusion of additional hyperedges. We hypothesize that an accurate modeling of concept dependencies is especially important for verbose natural

language queries. This is due to the fact that the grammatical complexity of these queries often challenges the capabilities of the current retrieval models (BENDERSKY and CROFT 2008; KUMARAN and CARVALHO 2009).

As an example, consider the verbose query used as a $\langle desc \rangle$ query in TREC topic §426:

“Provide information on the use of dogs worldwide for law enforcement purposes.”

Figure 8.1(a) shows an excerpt from the top document retrieved by a sequential dependence model (METZLER and CROFT 2005) – a state-of-the-art retrieval model that incorporates term dependencies – in response to this query. As evident from the excerpt in Figure 8.1(a), the top-retrieved document is *non*-relevant with respect to the query. Even though it contains many instances of the phrase “*law enforcement*” as well as the terms *provided* and *information* it does not mention the use of dogs.

On the other hand, an excerpt from the document in Figure 8.1(b) clearly indicates the relevance of the top document retrieved by our method with respect to the query. Even though this excerpt matches *less* of the query terms than the excerpt in Figure 8.1(a), it contains a relationship between the term *dog* and the phrase “*law enforcement*”, which is highly indicative of its relevance. This relationship cannot be modeled without accounting for higher-order term dependencies.

As Figure 8.1 shows, the evidence of the concepts co-occurring within a passage of text is a strong indicator of their dependency. This is somewhat akin to term dependencies, which are often modeled based on the frequency of the terms co-occurring next (or close) to each other in the document (METZLER and CROFT 2005; TAO and ZHAI 2007; LV and ZHAI 2009).

In the case of concept dependency, however, instead of relying on the entire document, we only examine a single document passage that is deemed to be the most relevant with respect to the query. This focused evidence can distinguish between

relevant documents and documents which simply contain many repeated concept instances, as in Figure 8.1(a). As we show in the next section, this approach is reminiscent of the passage retrieval models that often make use of the evidence from the highest-scoring document passage (BENDERSKY and KURLAND 2008; CALLAN 1994; CAI *et al.* 2004; KASZKIEL and ZOBEL 1997; WILKINSON 1994).

In contrast to the approach presented in this chapter, most passage retrieval methods are based on a conjunctive retrieval model and treat a query as a bag of words. However, as the excerpts in Figure 3.2 demonstrate, such a simple conjunctive retrieval model is not sufficient, especially for verbose, natural language queries.

Instead, the proposed retrieval framework distinguishes between the concepts and the dependencies that are crucial for conveying the query intent, and the concepts and the dependencies of lesser importance. For instance, in the case of the query in Figure 8.1, the dependency (*dog*, “*law enforcement*”) in Figure 8.1(b) is crucial for expressing the query intent, while the dependency (*information* and “*law enforcement*”) in Figure 8.1(a) is not.

To summarize, unlike any of the current retrieval models, the retrieval framework proposed in this chapter integrates three main characteristics that we believe are crucial for improving the effectiveness of retrieval with verbose queries. First, it models arbitrary term dependencies as concepts. Second, it uses passage-level evidence to model the dependencies between these concepts. Finally, it assigns weights to both concepts and concept dependencies, proportionate to the estimate of their importance for expressing the query intent. In this chapter, we show that by integrating these characteristics, the proposed retrieval framework can significantly improve the effectiveness of several current state-of-the-art retrieval models.

As in the rest of this dissertation, the proposed retrieval framework is based on a query representation using a *hypergraph* structure – a generalization of a graph, where an edge can connect more than two vertices. A vertex in a query hypergraph

corresponds to an individual query concept. The vertices are grouped by structures, which model various linguistic phenomena. For instance, a structure can group together terms, n-grams or noun phrases. Finally, any subset (rather than just a pair as in a standard graph) of vertices can be connected via a *hyperedge*, which models concept dependencies.

In this chapter, we use a query hypergraph representation that includes a *global hyperedge* to derive a ranking function that incorporates concepts and concept dependencies in a principled manner, based on the factorization of the hypergraph. We then derive several possible instantiations of this query hypergraph, which incorporate different structures and parameterization approaches.

The remainder of this chapter is organized as follows. Parameterized concept dependencies are inspired by passage-based retrieval methods that are described in Section 8.2. In Section 8.3, we show how parameterized concept dependencies can be modeled using query hypergraphs. In Section 8.4, we describe the optimization process of query hypergraphs with concept dependencies. In Section 8.5, we conduct retrieval experiments to demonstrate the superiority of the retrieval models that integrate concept dependencies to their counterparts that treat the query concepts independently. We conclude the chapter in Section 8.6.

8.2 Passages in Information Retrieval

The most commonly used form of information retrieval is *document retrieval*, i.e., retrieving an entire document (e.g., a news article or a web page) in response to a search query. However, there are some potential cases in which using only the most relevant document portions may be of value. These document portions are commonly referred to as *passages* in information retrieval (BENDERSKY and KURLAND 2008; LIU and CROFT 2002; CALLAN 1994).

Passages can be used in two ways in information retrieval. First, we can return the passages themselves in response to the search query. Alternatively, passages can be used to retrieve documents. In both cases, the retrieval task is to find passages that might pertain to a query. In the second case, however, these passages are used to evaluate the relevance of their ambient documents. In this section, we focus on the second case.

Passage-based evidence can be beneficial in information retrieval applications in several cases. First, it can be useful when only a small portion of a relevant document contains information that is actually relevant to the query. For example, consider a comprehensive book on the topic of information retrieval, wherein only a single section discusses passage-based retrieval. If the entire book is considered as an indivisible monolithic document, this section will have very limited influence on the overall document relevance score for a search query discussing the subject of passage-based retrieval.

Second, passage-based evidence can discover dependencies between the query concepts that go beyond exact phrases or proximity matches. For instance, consider the case of the query in Figure 8.1. While the concept pair (*law enforcement, dogs*) cannot be exactly matched as a phrase in the top-retrieved document in Figure 8.1(b), the fact that the two concepts co-occur within the confines of a passage that has a high query relevance score serves as important evidence of document relevance. This can especially benefit verbose queries that often contain concept dependencies that go beyond sequential phrases.

The main challenges in using passage-based evidence in document retrieval are (a) the identification of passage boundaries, and (b) the integration of passage evidence in the retrieval model. In the remainder of this section, we will discuss these two challenges.

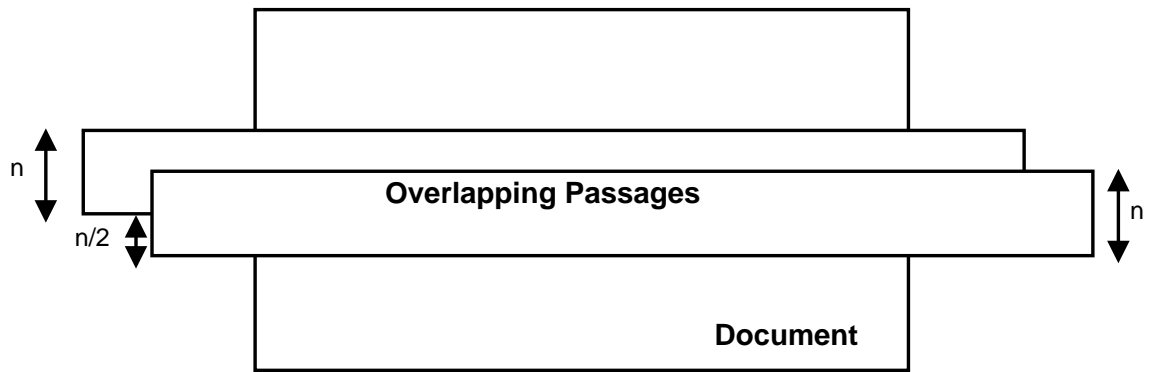


Figure 8.2. Overlapping passage identification.

8.2.1 Passage Identification

Passage types can be roughly classified into three main groups (CALLAN 1994; KASZKIEL and ZOBEL 2001): *discourse passages*, *semantic passages* and *window passages*.

Discourse passages are based on the document markup; examples include sentences, paragraphs or sections boundaries. Discourse passages have been found to work well for highly structured and edited corpora with clearly defined boundaries (CAI *et al.* 2004). However, in more heterogeneous collections, discourse passages do not contribute to consistent improvements in retrieval performance (CALLAN 1994).

Semantic passages are based on shifts of topic within a document. One of the most well known techniques to derive semantic passages is TextTiling (HEARST 1997). TextTiling groups adjacent blocks of text with high similarity into passages. Blocks are derived from sentence punctuation, and the similarity measure is the cosine similarity between the vector-space representation of pairs of adjacent blocks.

Window passages are passages that are based on fixed (or variable) number of words. This simple passaging technique was shown in some cases to be at least as effective as other techniques for passage identification for document retrieval (CALLAN 1994; KASZKIEL and ZOBEL 1997). This can be explained by the fact that semantic

passages may be hard to reliably identify in heterogeneous corpora (KASZKIEL and ZOBEL 1997).

A possible problem with dividing text into disjoint windows is that a small block of relevant text may be split between two passages. To overcome this problem overlapping windows are often used (CALLAN 1994). CALLAN (1994) and LIU and CROFT (2004) propose the following approach for building overlapping windows: begin the first passage in the beginning of the document, and create a new passage of length n every $\frac{n}{2}$ words. This *overlapping passages* approach is illustrated in Figure 8.2. Since overlapping passages were shown to be quite effective in previous work (LIU and CROFT 2002; BENDERSKY and KURLAND 2008), we adopt it as the passage identification method in this dissertation.

8.2.2 Passage-Based Retrieval Models

The most common way to integrate the passage-based evidence in the retrieval model is to combine the relevance score of the entire document with that of its passages. Since most of the current passage-based retrieval models are bag-of-words models, they can be expressed using the following equation

$$sc(Q, D) = \alpha \sum_{q \in Q} f(q, D) + (1 - \alpha) \mathcal{G}_{\pi \in \Pi_D} \left(\sum_{q \in Q} f(q, \pi) \right),$$

where Π_D is a set of passages derived from the document D using one of the passage identification techniques described in Section 8.2.1, \mathcal{G} is an arbitrary aggregation function, and $0 \leq \alpha \leq 1$ is a free parameter.

While, in theory, it is possible to use any arbitrary function \mathcal{G} to aggregate passage evidence, the most commonly used aggregation function is **max**. This aggregation approach, denoted **Max-Psg**, has been consistently shown to be successful in previous

work (CALLAN 1994; BENDERSKY and KURLAND 2008; WILKINSON 1994). Using the **max** aggregation function, we can rewrite the equation above as

$$sC_{\text{Max-Psg}}(Q, D) = \alpha \sum_{q \in Q} f(q, D) + (1 - \alpha) \max_{\pi \in \Pi_D} \left(\sum_{q \in Q} f(q, \pi) \right). \quad (8.1)$$

The success of the **Max-Psg** approach can be explained by the fact that it is designed to increase the score of documents that contain *at least one* very relevant passage to the query. In the context of information retrieval with verbose queries, the **Max-Psg** method can help to distinguish between *relevant* documents that contain several important concept dependencies within the confines of a single passage and the *non-relevant* documents that match many of the query terms scattered throughout the entire document (as in the case of the query in Figure 3.2).

In the next section, we demonstrate that the **Max-Psg** approach can be adopted to model concept dependencies in a query hypergraph. This gives rise to a retrieval model that – unlike the standard bag-of-words passage-based retrieval models (CALLAN 1994; BENDERSKY and KURLAND 2008; WILKINSON 1994; LIU and CROFT 2002) – can express arbitrary weighted concept dependencies.

8.3 Modeling Concept Dependencies with Query Hypergraphs

The **Max-Psg** retrieval method described in the previous section, can be viewed as a special case of the general query representation using query hypergraphs described in Chapter 3. Recall that the query hypergraph can model both concepts (i.e., term dependencies) as well as concept dependencies (i.e., higher-order term dependencies). The concepts are the vertices in the query hypergraph and the concept dependencies are the hyperedges (refer to Chapter 3 for more details on the query hypergraph induction process).

A convenient way to visually illustrate the concept dependencies in the query hypergraph is via a bipartite graph such as the one depicted in Figure 8.3. On the

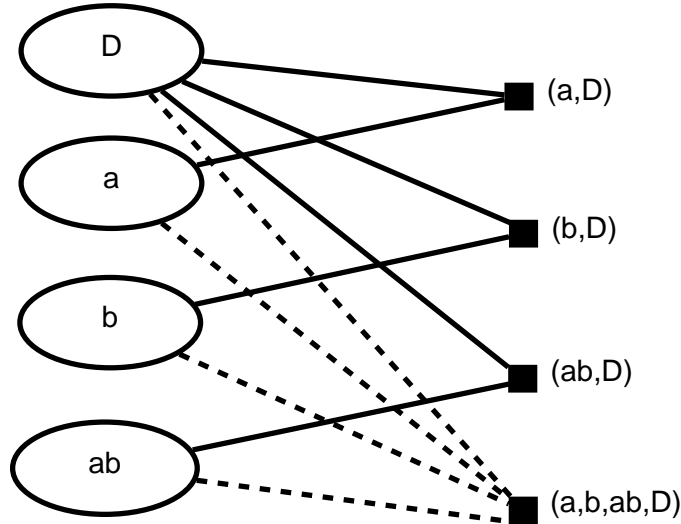


Figure 8.3. Bipartite graph representation of concept dependencies in a query hypergraph H . Local edges are represented by the solid edges in the bipartite graph. The global hyperedge is represented by the dashed edges in the bipartite graph.

left side of the graph, are the query concepts and the document (i.e., the hypergraph vertices). On the right side of the graph are the concept dependencies (i.e., the hyperedges). For instance, in the case of the query depicted in Figure 8.3, the following concept dependencies (or hyperedges) are modeled:

$$(a, D), (b, D), (ab, D), (a, b, ab, D).$$

Note that the document vertex is always included in a hyperedge, since we are interested in using the concept dependencies within the retrieval model, which assigns a relevance score to a document in response to the user query.

Recall from Section 3.2 that every hyperedge e in the query hypergraph H is associated with a factor $\phi_e(\mathbf{k}, D)$, which assigns a score to a dependency between a subset of concepts \mathbf{k}_e in the context of document D . Therefore, according to Equation 3.2, a relevance score of document D in response to query Q is given by the factorization of the query hypergraph H :

$$sc(Q, D) \triangleq \sum_{e \in E} \log(\phi_e(\mathbf{k}_e, D)).$$

Also, recall from the Section 3.3 that we consider two types of hyperedges (and the associated factors) in the query hypergraph: the *local* edges and the *global* hyperedge.

- The local edges are defined over the (*concept, document*) pairs. Examples of local edges are the concept dependencies (a, D) , (b, D) , (ab, D) in Figure 8.3. A local factor associated with a local edge is defined as

$$\phi(\{\kappa\}, D) \triangleq \exp\left(\lambda(\kappa)f(\kappa, D)\right),$$

where $\lambda(\kappa)$ is an importance weight assigned to the concept κ , and $f(\kappa, D)$ is a matching function between the concept κ and the document D . Refer to Section 3.3.3.1 for more details about the local factors.

- The global hyperedge $(\boldsymbol{\kappa}^Q, D)$ represents a dependency between the entire set of query concepts. Similarly to the **Max-Psg** retrieval model, the global factor uses a passage π , which receives the highest score among the set Π_D of passages extracted from the document D . Formally,

$$\phi(\boldsymbol{\kappa}^Q, D) \triangleq \exp\left(\max_{\pi \in \Pi_D} \sum_{\kappa \in \boldsymbol{\kappa}^Q} \lambda(\kappa, \boldsymbol{\kappa}^Q) f(\kappa, \pi)\right),$$

where $\lambda(\kappa, \boldsymbol{\kappa}^Q)$ is the importance weight of the concept κ in the context of the entire set of query concepts $\boldsymbol{\kappa}^Q$, and $f(\kappa, \pi)$ is a matching function between the concept κ and a passage $\pi \in \Pi_D$. Refer to Section 3.3.3.2 for more details about the global factor derivation.

Similarly to the **Max-Psg** method, the global factor assigns a higher relevance score to documents that contain a single highly-relevant passage. However, it is important to note that the query hypergraph representation with the global hyperedges

GlobalEdgeOptimization(Λ_L^0)

- 1: $\Lambda_G^0 \leftarrow \{0\}$
- 2: $\langle \mathcal{M}, \Lambda_L \rangle \leftarrow \mathbf{CoordinateAscent}(\emptyset, \Lambda_L^0)$
- 3: $\langle \mathcal{M}, \Lambda_G \rangle \leftarrow \mathbf{CoordinateAscent}(\Lambda_L, \Lambda_G^0)$
- 4: **return** $\langle \mathcal{M}, \Lambda_L \cup \Lambda_G \rangle$

Figure 8.4. Pipeline optimization of the parameterized query hypergraph with a global hyperedge.

has several important advantages compared to the standard bag-of-words **Max-Psg** formulations (CALLAN 1994; BENDERSKY and KURLAND 2008; WILKINSON 1994; LIU and CROFT 2002).

First, query hypergraphs can model passage-level dependencies between arbitrary concepts rather than single terms. This includes modeling a dependency between a phrase-term pair such as (*law enforcement, dogs*), which is impossible to model in the current bag-of-words **Max-Psg** formulations.

Second, query hypergraphs incorporate parameterized concept weighting based on a set of importance features. These parameterized weights can be assigned both to single concepts independently (as is done in the case of the local edges), and in the context of their co-occurrence with the other query concepts (as in the case of the global hyperedge).

Finally, the query hypergraph representation provides a principled method for optimizing the parameters of the concept weights in the local and the global hyperedges based on some specified retrieval metric \mathcal{M} . The specifics of this optimization are described in the next section.

8.4 Parameter Optimization

The parameterized concept weighting and expansion models using query hypergraphs that we considered thus far did not incorporate the global hyperedge. For instance, in the setting of the weighted sequential dependence model in Chapter 5,

we considered only the local edges connecting the concepts in the set of structures $\{\text{QT,PH,PR}\}$ with the document D . In the setting of the query expansion models in Chapter 6 and Chapter 7 we also considered the local edges connecting the expansion terms structure to the document D .

Adding the global hyperedge (and the associated factor $\phi_e(\kappa^Q, D)$) requires an additional optimization stage, as described in the pipeline optimization algorithm in Figure 8.4. First, the parameters associated with the local edges are optimized using the coordinate ascent method (line 2 of the algorithm).

Note that in the case of the query expansion methods, described in Section 6 and Section 7, the line 2 of the algorithm becomes a pipeline optimization instead (since both the explicit concept weights and the expansion term weights have to be optimized). However, in this dissertation, we only consider the application of query hypergraphs containing a global hyperedge to *non-expanded* queries. We leave the exploration of the application of query hypergraphs containing a global hyperedge in the query expansion methods to future work.

After the weights of the local factors are optimized, a second round of coordinate ascent optimization is performed. This time the parameters associated with the global factor are optimized.

Note that all the initial parameters associated with the global factor are set to zero. In such a way, we ensure that if the concept dependencies captured by the global hyperedge are not helpful in improving the retrieval performance (as measured by some retrieval metric \mathcal{M}), the global hyperedge will not be considered in the query hypergraph construction. Conversely, non-zero weights assigned to the global factor indicate that modeling concept dependencies via passage co-occurrence is beneficial for retrieval effectiveness.

<i>Retrieval Method</i>	QT	PH	PR	<i>Global Hyperedge</i>
QL	\mathcal{S}	–	–	–
\mathcal{H} -QL	\mathcal{S}	–	–	\mathcal{S}
SD	\mathcal{S}	\mathcal{S}	\mathcal{S}	–
\mathcal{H} -SD	\mathcal{S}	\mathcal{S}	\mathcal{S}	\mathcal{S}
FD	\mathcal{S}	\mathcal{S}	\mathcal{S}	–
\mathcal{H} -FD	\mathcal{S}	\mathcal{S}	\mathcal{S}	\mathcal{S}
			(+term subsets)	
			(+term subsets)	
WSD	\mathcal{C}	\mathcal{C}	\mathcal{C}	–
\mathcal{H} -WSD	\mathcal{C}	\mathcal{C}	\mathcal{C}	\mathcal{C}

Table 8.1. Retrieval baselines and their respective query hypergraph representation including the global hyperedge. \mathcal{S} indicates parameterization by structure, \mathcal{C} indicates parameterization by concept.

8.5 Evaluation

In this section, we compare the performance of the retrieval with query hypergraphs containing the global hyperedge to a number of state-of-the-art baselines that incorporate exact phrase matches, proximities, and concept weight parameterization. These baselines do not, however, incorporate concept dependencies.

The query hypergraph representation, proposed in this chapter, further extends each of these baselines with higher-order term dependencies via the inclusion of the global hyperedge and the corresponding global factor $\phi(\boldsymbol{\kappa}^Q, D)$ (see Section 8.3). In the remainder of this section, we examine the improvements in the retrieval performance (or lack thereof) of these baselines when they are extended with the query hypergraph representation including the global hyperedge.

All the initial retrieval parameters in the experiments reported in this section are set to the default Indri values, which reflect the best-practice settings. The parameter optimization and the evaluation are done using 3-fold cross-validation. The statistical significance of the differences in the performance of the retrieval methods is determined using a Fisher’s randomized test with 10,000 iterations and $\alpha < 0.05$.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
QL	33.09	24.24	47.62	25.66	23.85	12.75
\mathcal{H} -QL	34.12_q	25.49_q	49.13_q	27.24_q	24.1	13.07_q

(a) Query likelihood (QL) and its hypergraph representation (\mathcal{H} -QL).

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
SD	35.04	25.62	51.11	27.97	22.97	12.99
\mathcal{H} -SD	35.86_s	26.65_s	50.57	28.63_s	22.81	13.08

(b) Sequential dependence model (SD) and its hypergraph representation (\mathcal{H} -SD) parameterized by structure.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
FD	34.94	25.69	50.97	28.25	23.49	13.28
\mathcal{H} -FD	35.64_f	26.50_f	50.94	28.70_f	23.33	13.35

(c) Full dependence model (FD) and its hypergraph representation (\mathcal{H} -FD) parameterized by structure.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>	<i>P@20</i>	<i>MAP</i>
WSD	37.05	27.41	52.25	29.36	25.31	14.56
\mathcal{H} -WSD	37.07	27.79_w	51.68	29.82_w	25.57	14.68

(d) Weighted sequential dependence model (WSD) and its hypergraph representation (\mathcal{H} -WSD) parameterized by concept.

Table 8.2. Evaluation of the performance of the retrieval with query hypergraphs using binary metrics. Best result per column is marked in boldface. Statistically significant differences with a non-hypergraph baseline are marked by its title initial.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
QL	11.44	38.75	15.06	37.89	7.32	17.74
\mathcal{H} -QL	11.66	40.01_q	15.33	39.08_q	7.63	18.04

(a) Query likelihood (QL) and its hypergraph representation (\mathcal{H} -QL).

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
SD	11.76	40.91	15.73	40.97	7.58	17.11
\mathcal{H} -SD	11.93	41.95_s	15.93	40.7	7.78	17.44

(b) Sequential dependence model (SD) and its hypergraph representation (\mathcal{H} -SD) parameterized by structure.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
FD	11.87	40.82	16.10	40.94	8.21	18.02
\mathcal{H} -FD	11.94	41.65_f	16.02	41.01	8.15	17.92

(c) Full dependence model (FD) and its hypergraph representation (\mathcal{H} -FD) parameterized by structure.

	<i>Robust04</i>		<i>Gov2</i>		<i>ClueWeb-B</i>	
	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>	<i>ERR@20</i>	<i>NDCG@20</i>
WSD	12.04	42.86	16.52	42.47	8.58	19.58
\mathcal{H} -WSD	12.34_w	43.31	16.56	42.05	8.31	19.26

(d) Weighted sequential dependence model (WSD) and its hypergraph representation (\mathcal{H} -WSD) parameterized by concept.

Table 8.3. Evaluation of the performance of the retrieval with query hypergraphs using graded metrics. Best result per column is marked in boldface. Statistically significant differences with a non-hypergraph baseline are marked by its title initial.

We measure the performance using standard retrieval metrics for TREC corpora, as described in Section 4.2. For metrics that use binary relevance judgments, we use precision at the top 20 retrieved documents ($P@20$) and mean average precision across all the queries (MAP). For metrics that use graded relevance judgments, we use normalized discounted cumulative gain and expected reciprocal rank at rank 20 ($NDCG@20$ and $ERR@20$, respectively). We evaluate the retrieval methods under comparison using the three TREC corpora shown in Table 4.1.

Since the complex concept dependencies are most likely to benefit verbose queries, in this section we only report the retrieval effectiveness for the $\langle desc \rangle$ queries. Our preliminary experiments indicate that incorporating the global hyperedge does not result in significant effects on retrieval performance for the short $\langle title \rangle$ queries.

The main purpose of the empirical evaluation in this section is to examine the benefits that stem from adding a global hyperedge to a query hypergraph. To this end, we start with several *baseline query hypergraph representations* that incorporate a range of structures and have varying parameterizations, but do not include the global hyperedge. To each of these baselines representations, we add a global hyperedge. Thus for each baseline representation B , we create a hypergraph representation $\mathcal{H}\text{-}B$, which includes the global hyperedge.

Table 8.1 demonstrates these baselines and their respective representations including the global hyperedge. As we can see, Table 8.1 contains several hypergraphs that differ by the structures they contain and their parameterization. In the next sections, we examine the benefits that can be obtained by adding a global hyperedge to these baselines.

8.5.1 Comparison to the Query Likelihood Model

Query likelihood (PONTE and CROFT 1998) is a popular retrieval method that employs a bag-of-words query representation. In this section, we juxtapose the re-

trieval performance of the query likelihood baseline (denoted **QL**) to the performance of a query hypergraph that includes a single **QT**-structure (structure that contains the individual query terms as concepts) and the global hyperedge. We denote this hypergraph representation **H-QL**. This juxtaposition demonstrates the contribution of the global factor $\phi(\boldsymbol{\kappa}^Q, D)$ to the retrieval performance.

Table 8.2(a) and Table 8.3(a) demonstrate the comparison between the **QL** and the **H-QL** methods. The results in these tables show that the addition of the global factor $\phi(\boldsymbol{\kappa}^Q, D)$ into a bag-of-words representation significantly improves its retrieval effectiveness in all the cases, for both binary and graded retrieval metrics.

Note that the **H-QL** method is equivalent to the bag-of-words **Max-Psg** method that was shown to be effective in the previous work (BENDERSKY and KURLAND 2008; CAI *et al.* 2004; CALLAN 1994; KASZKIEL and ZOBEL 1997; WILKINSON 1994) and discussed in Section 8.2.2. **Max-Psg** ranks the documents in the collection by a combination of the document score and the score of its highest-scoring passage. Thus, the improvements in retrieval performance shown in Table 8.2(a) and Table 8.3(a) are in line with the improvements attained by the **Max-Psg** method reported in the previous work.

8.5.2 Comparison to the MRF-IR models

Markov random fields for information retrieval (MRF-IR) is a state-of-the-art retrieval framework that incorporates term dependencies. It was first proposed by METZLER and CROFT (2005), and was shown to be highly effective, especially for large-scale web collections.

METZLER and CROFT (2005) propose two instantiations of the general MRF-IR framework. The first instantiation is the sequential dependence model (denoted **SD**), which incorporates only dependencies between adjacent query terms. The second instantiation is the full dependence model (**FD**), which incorporates dependencies be-

tween all query term subsets. However, due to the verbosity of the description queries, in this paper, we limit our evaluation to query term subsets with at most three terms.

The SD and FD baselines can be represented with respective hypergraphs that include the structures QT, PR and PH, and only local edges. Both of these hypergraphs can be extended with a global hyperedge. We denote these extended hypergraph representations \mathcal{H} -SD and \mathcal{H} -FD, respectively. These hypergraphs are parameterized by structure, and their ranking functions are derived according to Equation 3.6.

Table 8.2(b) and Table 8.3(b) compare the performance of the sequential dependence baseline (SD) and its corresponding hypergraph \mathcal{H} -SD. As evident from these tables, in the majority of the cases the retrieval effectiveness (especially in terms of *MAP*) is significantly improved by the inclusion of the global hyperedge. However, these improvements are smaller than in the case of the QL baseline.

Similarly, Table 8.2(c) and Table 8.3(c) compare the performance of the full dependence baseline (FD) and its corresponding hypergraph \mathcal{H} -FD. Comparing the SD and the FD baselines, we can see that in most cases the FD baseline slightly outperforms the SD baseline. However, these differences were not found to be statistically significant.

When comparing the performance of the FD baseline and its corresponding hypergraph \mathcal{H} -FD, Table 8.2(c) and Table 8.3(c) demonstrate that the inclusion of the global factor results in an improved retrieval effectiveness (in terms of *MAP*) for all collections, and in statistically significant improvements for the *Robust04* and *Gov2* collections.

In addition, we can compare between the retrieval performance of the hypergraphs \mathcal{H} -SD and \mathcal{H} -FD. Similarly to the case of the baselines SD and FD, no statistically significant differences were found in the performance of these hypergraphs that include a global hyperedge. \mathcal{H} -FD is slightly more effective for the *ClueWeb-B* and the *Gov2* collections, while being slightly less effective for the *Robust04* collection.

8.5.3 Comparison to the Weighted Sequential Dependence Model

A major drawback of the SD and the FD baselines is that they are parameterized by structure, which ties the importance weights $\lambda(\cdot)$ of all the concepts that belong to the same structure (i.e., all the terms, phrases and proximities get the same respective weights). As shown in the experiments in Chapter 5, this parameterization can be detrimental, especially for longer, more verbose queries that may mix concepts of differing importance.

Recall that in Chapter 5 we proposed a weighted variant of the sequential dependence model (denoted WSD) that overcomes this drawback. The concept weights in the WSD method are parameterized using a set of importance features, associated with each concept based on its respective structure, as described in Chapter 5.

We extend the WSD baseline with a query hypergraph \mathcal{H} -WSD. The \mathcal{H} -WSD includes the global factor $\phi(\kappa^Q, D)$, which is also parameterized by concept. The ranking function for the \mathcal{H} -WSD hypergraph is presented in Equation 3.7.

Table 8.2(d) and Table 8.3(d) compare the retrieval performance of the WSD baseline and its corresponding hypergraph \mathcal{H} -WSD. While the retrieval improvements that stem from this hypergraph extensions are not as pronounced as in the cases of the QL, SD and FD baselines, the addition of the global factor to the WSD baseline still results in effectiveness gains for all the collections and most of the metrics.

For instance, for the *Gov2* collection, the \mathcal{H} -WSD method improves the performance (in terms of *MAP*) for 60% of the queries compared to the WSD baseline, while hurting only 30% of the queries. For 7% of the queries *MAP* is improved by more than 25%, while there is a 25% drop in performance for only 2% of the queries.

8.5.4 Further Retrieval Performance Analysis

In addition to comparing each individual query hypergraph model to its respective baseline in Table 8.1, some general trends can be observed in Table 8.2 and Table 8.3.

First, it is interesting to compare the relative differences in gains across the baselines, when the global hyperedge is added. The gains are the largest for the QL baseline, which does not include any term dependencies, and decrease as more term dependencies are added by the SD and the FD baselines. As an example, for the *Gov2* collection, the effectiveness gain as a result of the global factor inclusion decreases from 6.2% for the QL baseline to 1.6% for the FD baseline.

These diminishing returns demonstrate that there is some degree of overlap between the effect of term dependencies and higher-order term dependencies on the retrieval effectiveness. The overlap is not complete, however, since the addition of the global factor still has a statistically significant impact on the retrieval performance in most cases. This is true even for the FD baseline, which includes term dependencies between all query term pairs and triples.

Finally, we note that the parameterization of the ranking function by concept (as in the WSD baseline) (a) significantly improves the retrieval performance of the ranking function parameterized by structure (as in the SD baseline), and (b) further diminishes the gains obtained through the inclusion of the global factor. While \mathcal{H} -WSD is the best-performing retrieval method (in terms of *MAP*) in Table 8.2, its average effectiveness gain over the WSD baseline is only 1.3%. For comparison, the average effectiveness gain of the \mathcal{H} -QL method over the QL baseline is 4.7%.

8.5.5 Parameterization Analysis

In this section we analyze the parameterization of query hypergraphs. We examine both parameterization-by-structure and parameterization-by-concept regimes, which are described in detail in Section 3.4.1 and Section 3.4.2, respectively.

Recall that the parameters of the query hypergraph are optimized using the coordinate ascent algorithm such that the ranking function is decomposed into local and global factors (see Section 8.4). In this section, we display the resulting param-

eterization for the *Robust04* collection. We choose this collection, since it has the largest number of queries, and the learned parameterization is stable across all folds. However, it is important to note that the findings in this section hold for the other two collections as well.

8.5.5.1 Parameterization by Structure

Table 8.4 shows the hypergraph parameters for the local factors ($\lambda(\boldsymbol{\sigma})$) and the global factor ($\lambda(\boldsymbol{\sigma}, \Sigma^Q)$), averaged across folds, when the parameterization-by-structure approach is used (see Equation 3.6). These parameters correspond to the \mathcal{H} -SD model, the results for which are shown in Table 8.2(b) and Table 8.3(b).

Note that both for the local and the global factors the weights assigned to the term structure (QT) are the highest, which is in line with other models that incorporate term dependencies (METZLER and CROFT 2005). This demonstrates that despite the importance of term dependencies, individual term occurrences are still the most important indicators of relevance.

In addition, in Table 8.4, the parameters of the local factors are weighted higher than the parameters of the global factor. Recall that the global factor is defined over the highest-scoring passage in the document. Thus, the lower weight of the global factor parameters is in line with previous work, where passage evidence is typically weighted lower than the document evidence (BENDERSKY and KURLAND 2008; WILKINSON 1994; KASZKIEL and ZOBEL 1997).

Finally, note the *negative* weight assigned to the proximity (PR) structure in the global factor. While small, this negative weight is consistent across folds, as well as in the other collections. Intuitively, this negative weight indicates that in the highest-scoring passage of the relevant document we expect to encounter exact phrase concepts, rather than unordered proximity concepts.

	$\lambda(\boldsymbol{\sigma})$	$\lambda(\boldsymbol{\sigma}, \Sigma^Q)$
QT	+0.520	+0.322
PH	+0.065	+0.017
PR	+0.065	-0.011

Table 8.4. Query hypergraph parameterization by structure (*Robust04* collection).

φ	$\lambda(\varphi, \boldsymbol{\sigma})$		$\lambda(\varphi, \boldsymbol{\sigma}, \Sigma^Q)$	
	QT	PR+PH	QT	PR+PH
GF	-0.007	0	-0.005	-0.001
WF	+0.017	+0.007	+0.002	+0.002
QF	+0.012	0	+0.007	+0.008
CF	-0.021	0	-0.008	0
DF	-0.018	0	-0.001	0
AP	+0.540	+0.029	+0.298	+0.003

Table 8.5. Query hypergraph parameterization by concept (*Robust04* collection).

8.5.5.2 Parameterization by Concept

Table 8.5 shows the hypergraph parameters for the local factors ($\lambda(\varphi, \boldsymbol{\sigma})$) and the global factor ($\lambda(\varphi, \boldsymbol{\sigma}, \Sigma^Q)$), averaged across folds, when the parameterization-by-concept approach is used (see Equation 3.7). These parameters correspond to the \mathcal{H} -WSD model, the results for which are shown in Table 8.2(d) and Table 8.3(d). For the convenience of presentation and to reduce weight sparsity, we combine the weights of the PH and PR structures in the PR+PH column.

Note that the a priori constant importance feature AP generally receives the highest weight. This is due to the fact that setting all the other feature weights to zero yields exactly the parameterization-by-structure approach.

Features such as document frequency (DF), collection frequency (CF) and Google frequency (GF) receive, as expected, negative weights in most cases. In contrast, the query frequency (QF) and the Wikipedia title frequency (WF) features get positive weights, which indicates that the appearance of the concept in page title or in a search query is positively correlated to the concept importance.

(a) *What is the effect of Turkish river control projects on Iraqi water resources?*

Local Factor Weights

(0.0315 effect) (0.0451 turkish) (0.0508 river) (0.0313 control)
(0.0263 projects) (0.0413 iraqi) (0.0387 water) (0.0344 resources)
(0.0079 “effect turkish”) (0.0079 “turkish river”) (0.0096 “river control”)
(0.0079 “control projects”) (0.0079 “projects iraqi”)
(0.0079 “iraqi water”) (0.0194 “water resources”)

Global Factor Weights

(0.0203 effect 0.0262 turkish 0.0284 river 0.0164 control
0.0248 projects 0.0255 iraqi 0.0266 water 0.0252 resources
0.0014 “effect turkish” 0.0014 “turkish river” 0.0011 “river control”
0.0014 “control projects” 0.0014 “projects iraqi”
0.0014 “iraqi water” -0.0007 “water resources”)

(b) *What counterfeiting of money is being done in modern times?*

Local Factor Weights

(0.0610 counterfeiting) (0.0499 money)
(0.0408 done) (0.0614 modern) (0.0422 times)
(0.0178 “counterfeiting money”) (0.0178 “money done”)
(0.0178 “done modern”) (0.0468 “modern times”)

Global Factor Weights

(0.0198 counterfeiting 0.0067 money
0.0101 done 0.0105 modern 0.0039 times
0.0012 “counterfeiting money” 0 “money done”
0 “done modern” 0.0048 “modern times”)

Table 8.6. Examples of weights assigned to the concepts in the local and global factors.

8.5.5.3 Parameterization Examples

Table 8.6 demonstrates a full parameterization for two $\langle desc \rangle$ queries. This parameterization includes both the local factor weights and the global factor weights. In both cases, the parameterization by concept approach is used.

Table 8.6 illustrates some of the similarities and the differences between the independent concept weights (local factor weights) and their weights dependent on the other concepts (global factor weights). For instance, the weights of the single terms in both local and global factors in Table 8.6 follow roughly the same trend, while the phrase weights differ.

For query (a) in Table 8.6, the phrase *water resources*, which has the highest local factor weight, has a *negative* weight in the global factor. This setting corresponds to the intuition that a relevant *document* should contain the phrase *water resources*, however the most relevant *passage* in that document should focus on phrases that mention *Turkey* and *Iraq*.

Similar differences can be observed for query (b) in Table 8.6 as well. Note that some phrases in query (b) have zero weights in the global factor, while being assigned positive local factor weights.

8.6 Summary

In this chapter, we introduced query hypergraph representations that incorporate parameterized concept dependencies. Parameterized concept dependencies can model dependencies between arbitrary concepts in the query, rather than single query terms, and assign weights to these dependencies based on their contribution to the overall retrieval effectiveness of the query.

Parameterized concept dependencies are inspired by passage-based retrieval methods that are described in Section 8.2. In Section 8.3, we showed how parameterized concept dependencies can be modeled using query hypergraphs. In Section 8.4, we

described the optimization process of query hypergraphs with concept dependencies. In Section 8.5, we conducted retrieval experiments to demonstrate the superiority of the retrieval models that integrate concept dependencies to their counterparts that treat the query concepts independently.

This chapter concludes the exploration of various instantiations of the theoretical query hypergraph representation framework that we began in Chapter 5. We explored parameterized concept weighting in Chapter 5, parameterized query expansion in Chapter 6 and Chapter 7, and parameterized concept dependencies in this chapter. In the next – and final – chapter of this dissertation, we will summarize the findings of this dissertation and propose some potential directions for future work.

CHAPTER 9

SUMMARY AND FUTURE WORK

In this chapter we conclude this dissertation, and provide a broad perspective on our work. We start the chapter in Section 9.1 by highlighting the main steps in the process of query representation and information retrieval using query hypergraphs. Then, in Section 9.2 we summarize the key experimental results reported in this dissertation. We conclude the chapter and the dissertation in Section 9.3, where we discuss potential directions for future research.

9.1 Overview of the Query Hypergraphs

As described in this dissertation, query hypergraphs can model a variety of linguistic phenomena including concept weighting, query expansion and concept dependencies. However, it is important to note that constructing a query hypergraph representation requires only a handful of well-defined basic steps that we highlight in this section.

- (a) **Structures** In order to create a hypergraph we need to decide on a set of linguistic structures over which a hypergraph is defined. Each structure consists of individual concepts such as terms or phrases. These concepts are modeled as vertices in the query hypergraph.
- (b) **Hyperedges** Once the structures are defined, we need to define dependencies between them. These dependencies are modeled as hyperedges in the query hypergraph.

- (c) **Factors** Once the hyperedges are determined, we define factors associated with each hyperedge. These factors determine the contribution of each hyperedge to the total relevance score of a given document.
- (d) **Parameterization** Hypergraph parameterization may take several forms. We may parameterize by structure – that is we can tie the weights of all the concepts in the same structure. On the other hand, we may parameterize by concept and assign varying concept weights, based on a set of features indicating their importance. As we show in this dissertation, parameterization by concept leads to improved retrieval performance, especially for verbose queries.
- (e) **Parameter Optimization** Once the parameterization is defined we need to select a technique for optimizing the hypergraph parameters. In this dissertation, we propose a pipeline optimization method that can be used to optimize the parameters of a query hypergraph in multiple stages. This is especially useful for the cases when the query hypergraph representation can be affected by previous optimization steps (e.g., in the case of query expansion).
- (f) **Ranking Function** The ranking function finalizes the process of hypergraph construction, and combines the parameterized factors (with the optimized parameters) into a single document relevance score.

9.2 Summary of the Experimental Results

In this section, we summarize some of the key experimental results presented in this dissertation. We divide the examined retrieval methods into methods that use only the original query, and methods that employ query expansion. Since the focus of this dissertation is on verbose queries, we report the results of our experiments on the $\langle desc \rangle$ queries in this section.

$\langle desc \rangle$	<i>Robust04</i>	<i>Gov2</i>	<i>ClueWeb-B</i>
QL	24.24	25.66	12.75
SD	25.62* (+5.7%)	27.97* (+9.0%)	12.99 (+1.9)
WSD	27.41* _† (+13.1%)	29.36* _† (+14.4%)	14.56* _† (+14.3%)
\mathcal{H} -WSD	27.79* _† (+14.7%)	29.82* _† (+16.2%)	14.68* _† (+15.2%)

Table 9.1. Retrieval effectiveness gains, as measured by MAP , of query hypergraph based retrieval models (WSD, \mathcal{H} -WSD) compared to the current state-of-the-art retrieval models (QL, SD). The numbers in the parentheses indicate the percentage of improvement in MAP over the QL baseline. Statistically significant improvements with respect to QL and SD are marked by * and †, respectively.

$\langle desc \rangle$	<i>Robust04</i>	<i>Gov2</i>	<i>ClueWeb-B</i>
LCE	28.32	30.34	14.09
PQE	29.23* (+3.2%)	31.35* (+3.3%)	15.02 (+6.6%)
MSE	30.68* (+8.3%)	31.10 (+2.5%)	15.23* (+8.1%)

Table 9.2. Retrieval effectiveness gains, as measured by MAP , of query hypergraph based retrieval models that incorporate query expansion (PQE, MSE) compared to the latent concept expansion model (LCE). The numbers in the parentheses indicate the percentage of improvement in MAP over the LCE baseline. Statistically significant improvements with respect to LCE is marked by *.

Table 9.1 demonstrates a summary of the retrieval methods that use only the original query. As we see from Table 9.1, the non-parameterized retrieval methods (QL and SD) are significantly inferior to the parameterized retrieval method based on the query hypergraph representation (WSD and \mathcal{H} -WSD). The best-performing method, overall, is \mathcal{H} -WSD, which combines both parameterized concept weighting and parameterized concept dependencies. \mathcal{H} -WSD attains a consistent improvement of 15% or more in the *MAP* metric, compared to the QL retrieval method across all the corpora.

Table 9.2 demonstrates a summary of retrieval methods that use both the original query and the expansion terms. Table 9.2 shows that the latent concept expansion, a state-of-the-art query expansion method, is always less effective than the parameterized query expansion using either the retrieval corpus alone (PQE) or using multiple information sources (MSE). These improvements are statistically significant in the majority of the cases and range between 3% and 8%.

Finally, it is important to note that while the comparison in this section is based on the verbose *<desc>* queries, which are the main focus of this dissertation, the query hypergraph representation is robust enough to handle retrieval with both short keyword queries and verbose queries. In fact, as tables in Section 5.4, Section 6.5 and Section 7.5 demonstrate, query hypergraphs usually result in significant retrieval effectiveness improvements for short *<title>* queries as well.

9.3 Future Work

In our opinion, query hypergraphs are an important advance in information retrieval research in general, and, in particular, in retrieval with verbose, grammatically complex queries. However, retrieval with verbose queries presents many difficult research challenges, many of which are not addressed in this dissertation. Next, we describe some of these challenges and directions for potential future research.

- (a) **Query Hypergraphs with Arbitrary Features.** In this dissertation, we focused on query hypergraphs that contain linguistic structures. Thus, a vertex in a query hypergraph was a single textual concept that could be matched within the retrieved document. However, many of the current retrieval systems such as commercial web search engines use features that go beyond textual matches for the purposes of document retrieval ranking. These features include (but are not limited to) link-based features such as PageRank (BRIN and PAGE 1998), document formatting and layout (BENDERSKY *et al.* 2011), document reading level (COLLINS-THOMPSON *et al.* 2011) and visitation patterns (RICHARDSON *et al.* 2006). Incorporating these features that go beyond textual matches into the existing query hypergraph representation is a promising direction for future work with many practical applications.
- (b) **Natural Language Processing and Query Hypergraphs.** The linguistic structures that are used in the query hypergraph representation described in this dissertation are very basic and do not go beyond bigram phrases and proximity matches. Despite their simplicity, these structures result in significant retrieval performance improvements. These improvements are due to parameterized concept weighting and concept dependencies that are employed in the query hypergraph representation. However, it would be interesting to examine whether adding more complex linguistic structures that can be detected using natural language processing to the query hypergraphs will result in further gains in retrieval effectiveness. Examples of such structures may include noun and verb phrases, named entities, parse trees and semantic roles.
- (c) **Efficient Retrieval with Query Hypergraphs.** The focus of this dissertation is on retrieval effectiveness rather than retrieval efficiency. However, it is important to note that query hypergraphs can be used, in addition to pro-

viding effective query representations, to improve retrieval efficiency. A recent example of such approach is work by WANG *et al.* (2010) that use parameterized concept weights to reduce query runtime by dropping the lowest-weighted concepts. Similarly to this prior work, both parameterized query expansion and parameterized concept dependencies can serve as a basis for the development of more efficient retrieval models.

BIBLIOGRAPHY

- AGRAWAL, R., S. GOLLAPUDI, A. HALVERSON, and S. IEONG, 2009 Diversifying search results. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 5–14.
- AMATI, G. and C. J. VAN RIJSBERGEN, 2002 Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* 20(4): 357–389.
- BAI, J., Y. CHANG, H. CUI, Z. ZHENG, G. SUN, and X. LI, 2008 Investigation of partial query proximity in web search. In *Proceedings of the International Conference on World Wide Web*, pp. 1183–1184.
- BARR, C., R. JONES, and M. REGELSON, 2008 The Linguistic Structure of English Web-Search Queries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1021–1030.
- BENDERSKY, M. and W. B. CROFT, 2008 Discovering key concepts in verbose queries. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 491–498.
- BENDERSKY, M. and W. B. CROFT, 2009 Analysis of Long Queries in a Large Scale Search Log. In *Proceedings of Workshop on Web Search Click Data*, pp. 8–14.
- BENDERSKY, M. and W. B. CROFT, 2012 Modeling Higher-Order Term Dependencies in Information Retrieval using Query Hypergraphs. In *Proceedings of the Annual ACM SIGIR Conference (To appear)*.
- BENDERSKY, M., W. B. CROFT, and Y. DIAO, 2011 Quality-biased ranking of web documents. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 95–104.
- BENDERSKY, M., W. B. CROFT, and D. A. SMITH, 2009 Two-Stage Query Segmentation for Information Retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 810–811.
- BENDERSKY, M., D. FISHER, and W. B. CROFT, 2011 UMass at TREC 2010 Web Track: Term Dependence, Spam Filtering and Quality Bias. In *Proceedings of TREC-10*.
- BENDERSKY, M. and O. KURLAND, 2008 Utilizing Passage-Based Language Models for Document Retrieval. In *Proceedings of the European Conference on Information Retrieval*, pp. 162–174.

- BENDERSKY, M., D. METZLER, and W. B. CROFT, 2010 Learning concept importance using a weighted dependence model. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 31–40.
- BENDERSKY, M., D. METZLER, and W. B. CROFT, 2011 Parameterized Concept Weighting in Verbose Queries. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 605–614.
- BENDERSKY, M., D. METZLER, and W. B. CROFT, 2012 Effective Query Formulation with Multiple Information Sources. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 443–452.
- BERGSMA, S. and Q. I. WANG, 2007 Learning Noun Phrase Query Segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 819–826.
- BISHOP, C. M., 2006 *Pattern Recognition and Machine Learning*. Springer.
- BRANTS, T. and A. FRANZ, 2006 Web 1T 5-gram Version 1.
- BRIN, S. and L. PAGE, 1998 The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30(1-7): 107–117.
- BRODER, A., 2002 A taxonomy of web search. *the Annual ACM SIGIR Conference Forum* 36(2): 3–10.
- BUCKLEY, C. and E. M. VOORHEES, 2004 Retrieval Evaluation with Incomplete Information. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 25–32.
- BURGES, C., T. SHAKED, E. RENSHAW, A. LAZIER, M. DEEDS, N. HAMILTON, and G. HULLENDER, 2005 Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine learning*, pp. 89–96.
- CAI, D., S. YU, J.-R. WEN, and W.-Y. MA, 2004 Block-based web search. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 456–463.
- CALLAN, J., 1994 Passage-level evidence in document retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 302–310.
- CAO, G., J.-Y. NIE, J. GAO, and S. ROBERTSON, 2008 Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 243–250.
- CHAPELLE, O., D. METZLER, Y. ZHANG, and P. GRINSPAN, 2009 Expected Reciprocal Rank for Graded Relevance. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 621–630.
- CLARKE, C. L., M. KOLLA, G. V. CORMACK, O. VECHTOMOVA, A. ASHKAN, S. BÜTTCHER, and I. MACKINNON, 2008 Novelty and diversity in information retrieval evaluation. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 659–666.
- CLARKE, C. L. A., N. CRASWELL, and I. SOBOROFF, 2010 Overview of the TREC 2009 Web Track. In *Proceedings of TREC-2009*.

- COLLINS-THOMPSON, K., P. N. BENNETT, R. W. WHITE, S. DE LA CHICA, and D. SONTAG, 2011 Personalizing web search results by reading level. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 403–412.
- COLLINS-THOMPSON, K. and J. CALLAN, 2005 Query expansion using random walk models. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 704–711.
- CRASWELL, N., O. ZOETER, M. TAYLOR, and B. RAMSEY, 2008 An experimental comparison of click position-bias models. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 87–94.
- CROFT, W. B., D. METZLER, and T. STROHMAN, 2009 *Search Engines: Information Retrieval in Practice*. Addison-Wesley.
- CUMMINS, R. and C. O’RIORDAN, 2009 Learning in a pairwise term-term proximity framework for information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 251–258.
- DANG, V. and W. B. CROFT, 2010 Feature Selection for Document Ranking Using Best First Search and Coordinate Ascent. In *the Annual ACM SIGIR Conference Workshop on Feature Generation and Selection for Information Retrieval*.
- DIAZ, F. and D. METZLER, 2006 Improving the estimation of relevance models using large external corpora. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 154–161.
- DOWNEY, D., S. DUMAIS, D. LIEBLING, and E. HORVITZ, 2008 Understanding the relationship between searchers’ queries and information goals. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 449–458.
- FAGAN, J., 1987 Automatic phrase indexing for document retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 91–101.
- FENG, J., M. JOHNSTON, and S. BANGALORE, 2011 Speech and Multimodal Interaction in Mobile Search. *Signal Processing Magazine, IEEE* 28(4): 40–49.
- FERRUCCI, D., E. BROWN, J. CHU-CARROLL, J. FAN, D. GONDEK, A. KALYANPUR, A. LALLY, J. MURDOCK, E. NYBERG, J. PRAGER, and OTHERS, 2010 Building Watson: An overview of the DeepQA project. *AI Magazine* 31(3): 59–79.
- FINKEL, J. R., 2010 *Holistic Language Processing: Joint Models of Linguistic Structure*. Ph. D. thesis, Stanford University.
- GAO, J., J.-Y. NIE, G. WU, and G. CAO, 2004 Dependence language model for information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 170–177.

- GAO, J., H. QI, X. XIA, and J.-Y. NIE, 2005 Linear discriminant model for information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 290–297.
- HEARST, M., 1997 TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics* 23(1): 33–64.
- HECHT, B., J. TEEVAN, M. R. MORRIS, and D. J. LIEBLING, 2012 SearchBuddies: Bringing search engines into the conversation. In *Proceedings of International AAAI Conference on Weblogs and Social Media*.
- HOROWITZ, D. and S. D. KAMVAR, 2010 The Anatomy of a Large Scale Social Search Engine. In *Proceedings of the International Conference on World Wide Web*, pp. 431–440.
- JÄRVELIN, K. and J. KEKÄLÄINEN, 2002 Cumulated gain-based evaluation of IR techniques. *ACM Transactions of Information Systems (TOIS)* 20(4): 422–446.
- JEON, J., W. B. CROFT, and J. H. LEE, 2005 Finding similar questions in large question and answer archives. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 84–90.
- JOACHIMS, T., 2002 Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 133–142.
- JONES, R. and K. L. KLINKNER, 2008 Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 699–708.
- KASZKIEL, M. and J. ZOBEL, 1997 Passage retrieval revisited. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 178–185.
- KASZKIEL, M. and J. ZOBEL, 2001 Effective ranking with arbitrary passages. *Journal of the American Society for Information Science* 52: 344–364.
- KAUFMANN, M., M. VAN KREVELD, and B. SPECKMANN, 2009 Subdivision Drawings of Hypergraphs. In I. Tollis and M. Patrignani (Eds.), *Graph Drawing*, Volume 5417 of *Lecture Notes in Computer Science*, Chapter 39, pp. 396–407. Springer Berlin / Heidelberg.
- KUMARAN, G. and V. R. CARVALHO, 2009 Reducing long queries using query quality predictors. In *Proceedings of the Annual ACM SIGIR Conference*, New York, NY, USA, pp. 564–571.
- KWOK, K. L., 1990 Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems (TOIS)* 8(4): 363–386.
- LANG, H., D. METZLER, B. WANG, and J.-T. LI, 2010 Improved latent concept expansion using hierarchical markov random fields. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 249–258.

- LAVRENKO, V. and B. W. CROFT, 2003 Relevance Models in Information Retrieval. In B. W. Croft and J. Lafferty (Eds.), *Language Modeling for Information Retrieval*, pp. 11–56. Kluwer.
- LEASE, M., 2009 An improved markov random field model for supporting verbose queries. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 476–483.
- LEASE, M., J. ALLAN, and W. B. CROFT, 2009 Regression Rank: Learning to Meet the Opportunity of Descriptive Queries. In *Proceedings of the European Conference on Information Retrieval*, pp. 90–101.
- LI, H., 2011 *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan and Claypool Publishers.
- LI, P., C. BURGESS, and Q. WU, 2007 Learning to rank using classification and gradient boosting. In *Proceedings of NIPS*.
- LIN, J., D. METZLER, T. ELSAYED, and L. WANG, 2010 Of Ivory and Smurfs: Loxodontan MapReduce Experiments for Web Search. In *Proceedings of TREC-09*.
- LIN, Y., H. LIN, S. JIN, and Z. YE, 2011 Social annotation in query expansion: a machine learning approach. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 405–414.
- LIU, X. and W. B. CROFT, 2002 Passage retrieval based on language models. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 375–382.
- LIU, X. and W. B. CROFT, 2004 Cluster-based retrieval using language models. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 186–193.
- LV, Y. and C. ZHAI, 2009 Positional language models for information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 299–306.
- LV, Y. and C. ZHAI, 2010 Positional relevance model for pseudo-relevance feedback. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 579–586.
- MCCREADIE, R., C. MACDONALD, I. OUNIS, J. PENG, and R. L. T. SANTOS, 2010 University of Glasgow at TREC 2009: Experiments with Terrier. In *Proceedings of TREC-09*.
- MEI, Q. and K. CHURCH, 2008 Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 45–54.
- METZLER, D., 2007a Using Gradient Descent to Optimize Language Modeling Smoothing Parameters. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 687–688.
- METZLER, D. and W. B. CROFT, 2005 A Markov random field model for term dependencies. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 472–479.

- METZLER, D. and W. B. CROFT, 2007a Latent concept expansion using markov random fields. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 311–318.
- METZLER, D. and W. B. CROFT, 2007b Linear Feature-Based Models for Information Retrieval. *Information Retrieval* 10(3): 257–274.
- METZLER, D. A., 2007b Automatic feature selection in the markov random field model for information retrieval. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 253–262.
- MISHNE, G. and M. DE RIJKE, 2005 Boosting Web Retrieval Through Query Operations. In *Proceedings of the European Conference on Information Retrieval*, pp. 502–516.
- MOHAN, A., Z. CHEN, and K. Q. WEINBERGER, 2011 Web-Search Ranking with Initialized Gradient Boosted Regression Trees. *Journal of Machine Learning Research, Workshop and Conference Proceedings* 14: 77–89.
- NALLAPATI, R. and J. ALLAN, 2002 Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 383–390.
- PARK, J. and W. B. CROFT, 2010 Query Term Ranking based on Dependency Parsing of Verbose Queries. In *Proceedings of the Annual International SIGIR Conference*, pp. 829–830.
- PARK, J. H., W. B. CROFT, and D. A. SMITH, 2011 A quasi-synchronous dependence model for information retrieval. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 17–26.
- PENG, J., C. MACDONALD, B. HE, V. PLACHOURAS, and I. OUNIS, 2007 Incorporating term dependency in the DFR framework. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 843–844.
- PONTE, J. M. and W. B. CROFT, 1998 A language modeling approach to information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 275–281.
- RICHARDSON, M., A. PRAKASH, and E. BRILL, 2006 Beyond PageRank: machine learning for static ranking. In *Proceedings of the International Conference on World Wide Web*, pp. 707–715.
- ROBERTSON, S. E. and K. SPARCK JONES, 1988 Relevance weighting of search terms. In P. Willett (Ed.), *Document retrieval systems*, pp. 143–160. London, UK, UK: Taylor Graham Publishing.
- ROBERTSON, S. E. and S. WALKER, 1994 Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 232–241.
- ROCCHIO, J., 1971 *Relevance Feedback in Information Retrieval*, pp. 313–323. Prentice Hall.

- SALTON, G. and C. BUCKLEY, 1988 Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5): 513–523.
- SALTON, G., A. WONG, and C. S. YANG, 1975 A vector space model for automatic indexing. *Communications of the ACM* 18(11): 613–620.
- SANTOS, R. L., C. MACDONALD, and I. OUNIS, 2010 Exploiting query reformulations for web search result diversification. In *Proceedings of the International Conference on World Wide Web*, pp. 881–890.
- SANTOS, R. L., C. MACDONALD, and I. OUNIS, 2011 Intent-aware search result diversification. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 595–604.
- SHI, L. and J.-Y. NIE, 2010 Using various term dependencies according to their utilities. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 1493–1496.
- SMUCKER, M. D. and J. ALLAN, 2006 Lightening the load of document smoothing for better language modeling retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 699–700.
- SMUCKER, M. D., J. ALLAN, and B. CARTERETTE, 2007 A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 623–632.
- SPARCK JONES, K., 1988 A statistical interpretation of term specificity and its application in retrieval. In P. Willett (Ed.), *Document retrieval systems*, pp. 132–142. Taylor Graham Publishing.
- STOCK, W., 2010 Concepts and semantic relations in information science. *Journal of the American Society for Information Science and Technology* 61(10): 1951–1969.
- STROHMAN, T., D. METZLER, H. TURTLE, and W. B. CROFT, 2004 Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*.
- SVORE, K. M., P. H. KANANI, and N. KHAN, 2010 How good is a span of terms?: exploiting proximity to improve web retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 154–161.
- TAN, B. and F. PENG, 2008 Unsupervised query segmentation using generative language models and Wikipedia. In *Proceedings of the International Conference on World Wide Web*, pp. 347–356.
- TAO, T. and C. ZHAI, 2007 An exploration of proximity measures in information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 295–302.
- TURTLE, H. and W. B. CROFT, 1991 Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems (TOIS)* 9(3): 187–222.

- WANG, L., D. METZLER, and J. LIN, 2010 Ranking under temporal constraints. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 79–88.
- WANG, M. and L. SI, 2008 Discriminative probabilistic models for passage based retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 419–426.
- WILKINSON, R., 1994 Effective retrieval of structured documents. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 311–317.
- XU, J. and W. B. CROFT, 1996 Query expansion using local and global document analysis. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 4–11.
- XU, J. and H. LI, 2007 AdaRank: a boosting algorithm for information retrieval. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 391–398.
- XU, Y., G. J. F. JONES, and B. WANG, 2009 Query dependent pseudo-relevance feedback based on Wikipedia. In *Proceedings of the Annual ACM SIGIR Conference*, pp. 59–66.
- YU, C. T., C. BUCKLEY, K. LAM, and G. SALTON, 1983 A Generalized Term Dependence Model in Information Retrieval. Technical report, Cornell University.
- ZHAI, C. and J. LAFFERTY, 2004 A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)* 22(2): 179–214.
- ZHAO, L. and J. CALLAN, 2010 Term Necessity Prediction. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pp. 43–52.
- ZOBEL, J. and A. MOFFAT, 1998 Exploring the similarity space. *SIGIR Forum* 32(1): 18–34.