

SEARCH USING SOCIAL MEDIA STRUCTURES

A Dissertation Presented

by

JANGWON SEO

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 2011

Computer Science

© Copyright by Jangwon Seo 2011

All Rights Reserved

SEARCH USING SOCIAL MEDIA STRUCTURES

A Dissertation Presented

by

JANGWON SEO

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

Andrew McCallum, Member

David A. Smith, Member

Weibo Gong, Member

Andrew G. Barto, Department Chair
Computer Science

To Miyong and my parents

ACKNOWLEDGMENTS

This work would not have been possible without the support from a number of people. Most importantly, I would like to thank my advisor, W. Bruce Croft. Working with him has been an exciting experience. He led me to focus on interesting topics and do meaningful research. Also, he encouraged me, giving me lots of freedom so that I could explore various areas in Information Retrieval and study interdisciplinary approaches.

I would also like to thank my thesis committee members James Allan, Andrew McCallum, David A. Smith and Weibo Gong for their helpful comments. Furthermore, I could greatly improve my understanding of Information Retrieval, Natural Language Processing and Graph Theory by taking their classes or having discussions with them.

I should also like to thank Jiwoon Jeon, Joon Ho Lee and Kyoungsoo Lee. Early discussions with them inspired me to focus on social media domain and develop novel representation techniques. I also thank my fellow CIIR graduate students and visitors.

Additionally, I would thank Fernando Diaz, Bo Pang, Evgeniy Gabrilovich and Vanja Josifovski for mentoring me during my summer at Yahoo! Research. Although my collaboration with them was not a part of this thesis, I benefited from interactions with them.

Finally, I want to recognize Miyong Ko, my wife, for her whole-hearted support and trust. She put in great efforts to make my life happy as a PhD student at University of Massachusetts Amherst and encouraged me every step of the way.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-0711348, in part by NSF CLUE IIS-0844226, and in part by NSF grant #IIS-0534383. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the sponsors.

ABSTRACT

SEARCH USING SOCIAL MEDIA STRUCTURES

SEPTEMBER 2011

JANGWON SEO

B.Sc., SEOUL NATIONAL UNIVERSITY, SEOUL, KOREA

M.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Social applications on the Web have appeared as communication spaces for sharing knowledge and information. In particular, social applications can be considered valuable information sources because information in the applications is not only easily accessible but also revealing in that the information accrues via interactions between people.

In this work, we address methods for finding relevant information in social media applications that use unique properties of these applications. In particular, we focus on three unique structures in social media: hierarchical structure, conversational structure, and social structure. Hierarchical structures are used to organize information according to certain rules. Conversational structures are formed by interactions within communities such as replies. Social structures represent social relationships among community members. These structures are designed to organize information and encourage people to participate in discussions in social applications. Accordingly,

contexts extracted from these structures can be used to improve the effectiveness of search in social media relative to representations based solely on text content.

To exploit these structures in retrieval frameworks, we need to address three challenges as follows. First, we should discover each structure because it is often obscure. Second, we need to extract relevant contexts from each structure because not all the contexts in a structure are relevant for retrieval. Last, we should represent each context or their combinations in a representation framework so that they can be encoded as retrieval components such as documents. In this work, we introduce an effective representation framework for multiple contexts. We then discuss how to discover or define each structure and how to extract relevant contexts from the structure. Using the representation framework, these relevant contexts are integrated into retrieval algorithms. To demonstrate that these structures can improve search in social media, the retrieval models and frameworks incorporating these structures are evaluated through experiments using data collections gathered from a variety of social media applications.

In addition, we address two minor challenges related to social media search. First, it is not always easy to find relevant information from relevant objects if the objects are large. Accordingly, we address identification of relevant substructures in such objects. Second, text reuse structures are important since these structures have the potential to affect various retrieval tasks. In this thesis, we introduce text reuse structures and analyze text reuse patterns in real social applications.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	vii
LIST OF TABLES	xiv
LIST OF FIGURES	xix
 CHAPTER	
1. INTRODUCTION	1
1.1 Structures in Social Media	2
1.2 Major Retrieval Challenges	6
1.3 Minor Challenges	8
1.4 Contributions	9
1.5 Organization	10
2. RELATED WORK	12
2.1 Hierarchical Structure	12
2.2 Conversational Structure	14
2.3 Social Structure	15
2.4 Our Published Work	16
3. GEOMETRIC REPRESENTATIONS FOR MULTIPLE CONTEXTS	18
3.1 Related Work	20
3.2 The log-linearity of the geometric mean	22
3.3 Geometry of Multiple Documents	22
3.3.1 Fréchet Mean	23
3.3.2 Euclidean Metric space	24

3.3.3	Riemannian manifold defined by the Fisher information metric	27
3.4	Experiments	29
3.4.1	Cluster Retrieval	30
3.4.2	Pseudo-Relevance Feedback	33
3.5	Conclusions	35
4.	HIERARCHICAL STRUCTURES AND BLOG SITE SEARCH	37
4.1	Blog Site Search	37
4.2	Blog Site Representations Using Hierarchical Contexts	38
4.2.1	Global Representation	40
4.2.2	Query Generation Maximization	41
4.2.3	Pseudo-Cluster based Selection	43
4.3	Experiments	45
4.3.1	Data	45
4.3.2	Experimental Design	46
4.3.3	Results	47
4.4	Incorporating Global Contexts	48
4.4.1	Types of Blog Sites	48
4.4.2	Diversity Penalty	51
4.4.2.1	Diversity Penalty by Global Representation	51
4.4.2.2	Clarity Score as a Penalty Factor	53
4.4.2.3	Diversity Penalty by Random Sampling	54
4.4.3	Experimental Results	56
4.5	Blog Distillation Task	58
4.6	Conclusion	61
5.	CONVERSATIONAL STRUCTURES AND ONLINE COMMUNITY SEARCH	62
5.1	Discovery of Thread Structure	63
5.1.1	Intrinsic Features	67
5.1.2	Extrinsic Features	69

5.1.3	Learning	73
5.1.4	Collections	74
5.1.5	Experiments	77
5.1.6	Results and Discussion	78
5.2	Multiple Context-based Retrieval	84
5.2.1	Context Extraction based on Thread Structure	84
5.2.2	Multi-context-based Retrieval	85
5.2.2.1	Thread Search	86
5.2.2.2	Posting Search	87
5.2.3	Test Collections	88
5.2.4	Experiments	90
5.2.5	Results	91
5.2.6	Comparison with cluster-based language model	93
5.3	Conclusions	94
6.	SOCIAL STRUCTURES AND EXPERT FINDING	96
6.1	Graph-based Expert Finding Techniques	97
6.1.1	Posting-based Graph Construction	98
6.1.2	Thread-based Graph Construction	98
6.1.3	Expertise Ranking	100
6.2	Experiments	102
6.2.1	Email Archive	102
6.2.2	Forum	104
6.3	Conclusions	106
7.	SEARCH USING THREE STRUCTURES	107
7.1	Representation Combining Three Structures	107
7.2	Experiments	111
7.2.1	Constructing a test collection by crowd-sourcing	113
7.2.2	Results	116
7.3	Conclusions	118

8. IDENTIFYING RELEVANT SUBSTRUCTURES	119
8.1 Related Work	120
8.2 Estimating Posting-level Relevance	120
8.2.1 Posting Query-likelihood	121
8.2.2 Multi-context Interpolation	121
8.2.3 Enhancement via Query Expansion	122
8.3 Relevance Maximization Through Thread Structures	123
8.3.1 Greedy Approach	124
8.3.2 Mixed Integer Programming Approach	125
8.4 Experiments	126
8.4.1 Forum Data	126
8.4.2 Evaluation and Baselines	127
8.4.3 Results	130
8.5 Conclusions	131
9. TEXT REUSE STRUCTURES AND TEXT REUSE PATTERN ANALYSIS	133
9.1 Related Work	134
9.2 Text Reuse Basics	135
9.2.1 Definitions of Text Reuse	135
9.2.2 Text Reuse Detection	137
9.3 Text Reuse Pattern Analysis in Blogs	138
9.3.1 DCT fingerprinting	138
9.3.2 Results and Discussions	141
9.4 Text Pattern Analysis in Microblogs	144
9.4.1 Locality Sensitive Hashing (LSH)	144
9.4.2 Pattern Analysis	145
9.5 Conclusions	147
10. CONCLUSIONS	149
10.1 Summaries of Chapters	149
10.2 Our Contributions	151

10.3 Future Work	153
APPENDICES	
A. GEOMETRY OF MULTIPLE DOCUMENTS	156
B. UNSUPERVISED ESTIMATION OF DIRICHLET SMOOTHING PARAMETERS	165
BIBLIOGRAPHY	171

LIST OF TABLES

Table	Page
3.1 Test collections.	30
3.2 Results for cluster retrieval. A-MEAN, G-MEAN and SELECT mean representations by the arithmetic mean, by the geometric mean, and by geometric selection, respectively. The numbers are P@5 scores. A * indicates a statistically significant improvement over A-MEAN ($p < 0.05$).	33
3.3 Results for pseudo-relevance feedback. RM and GRM mean the relevance model and the geometric relevance model, respectively. The numbers are MAP scores. A * indicates a statistically significant improvement over RM ($p < 0.01$).	35
4.1 The criteria for the relevance judgments.	46
4.2 Retrieval performance by blog site representation techniques. α and β in a cell indicate statistically significant improvement over the baselines, global representation and query generation maximization, respectively. ($p < 0.1$)	48
4.3 Type classification of blog sites	50
4.4 Manual classification result with 100 blog sites.	51
4.5 Retrieval performance for blog site representation techniques combined with each penalty factor. GR, QGM and PCS stand for global representation, query generation maximization and pseudo-cluster selection, respectively. α and β in a cell indicate statistically significant improvement ($p < 0.1$) over the baselines, global representation and query generation maximization, respectively.	56

4.6	Retrieval performance for the blog distillation task. GR, QGM and PCS stand for global representation, query generation maximization and pseudo-cluster selection, respectively. α and β in a cell indicate statistically significant improvement ($p < 0.1$) over the baselines, global representation and query generation maximization, respectively.	59
5.1	Statistics of collections	75
5.2	Thread structure discovery results on the WOW collection. Values are accuracy scores. Each row corresponds to an intrinsic feature: full text (F), original contents (O), quotations (Q), unigram (U) and n-gram (N). Each column corresponds to an extrinsic feature: location prior (LP), time gap (TG), author reference (AR), same author (SA), inferred turn-taking (IT) and all extrinsic features (ALL). Bold values indicate the best score group, i.e., the score is not statistically significantly different from the best score (by the paired randomization test with p -value < 0.05).	79
5.3	Thread structure discovery results on the Cancun collection	79
5.4	Thread structure discovery results on the W3C collection	79
5.5	Thread structure discovery accuracy on baselines. Two baselines (the first and second rows) consider specific thread structures, i.e., the top-based structure and the chronological structure. Another baseline (the third row) uses the graph-based propagation algorithm [26].	81
5.6	Example queries for the WOW collection and the Cancun collection	89
5.7	Summary of relevance judgments of two forum collections (WOW and CANCUN). The numbers of judged threads and relevant threads are averaged per topic.	89
5.8	Retrieval Performance on the WOW collection (Thread Search). The superscripts α , β and γ indicate statistically significant improvements on each baseline, i.e., ‘Thread’, ‘Posting’, ‘Posting + Thread’, respectively (by the paired randomization test with p -value < 0.05).	92
5.9	Retrieval Performance on the Cancun collection (Thread Search)	92

5.10	Retrieval performance of the WOW collection (based on inaccurate thread structure discovery)	92
5.11	Retrieval performance on the W3C collection (Posting Search). The superscripts α and β indicate statistically significant improvements on the baselines, i.e., ‘Posting’ and ‘Posting + Thread’, respectively (by the paired randomization test with p -value < 0.05)	93
5.12	Retrieval performance of cluster-based language models on the W3C collection (Posting Search). These results do not show statistically significant differences from the baseline ‘Posting’ in Table 5.11 (by the paired randomization test with p -value < 0.05).	94
6.1	Expert finding results for different graph construction methods on the W3C collection. ‘Posting’, ‘Thread’ and ‘Thread Structure’ represent the posting-based, thread-based, and thread structure-based graph construction methods, respectively. ($c \rightarrow p$) and ($p \rightarrow c$) mean the direction of child-to-parent and parent-to-child for posting-to-posting edges. Superscripts α and β indicate statistically significant improvements on ‘Posting’ and ‘Thread’, respectively. (the paired randomization test with p -value < 0.1)	104
6.2	Examples of the Apple Discussion forums used for the test collection	105
6.3	Expert finding results for different graph construction methods on the Apple forums. Superscripts α and β indicate statistically significant improvements on ‘Thread’ and ‘Thread Structure ($c \rightarrow p$)’, respectively. (the paired randomization test with p -value < 0.05)	106
7.1	Statistics of the Whiteblaze.net collection	114
7.2	Examples of queries for Whiteblaze.net	114
7.3	Results by the three structure combination on WOW and CANCUN. “Dialogue + Thread” is the best result from Chapter 5. A † indicates a statistically significant improvement on “Dialogue + Thread” (randomization test with p -value < 0.05).	117
7.4	Results by the three structure combination on Whiteblaze.net. A † indicates a statistically significant improvement on “Thread” (randomization test with p -value < 0.05).	117

8.1	Evaluation results of proposed techniques and baselines (Chronological order and Posting-level Relevance order) according to different cutoffs when query expansion is not employed. “MIP” denotes the mixed integer programming approach. A bold number indicates the best performance for each cutoff. Since the number of the corresponding topics depending on the cutoff varies, the number are also reported.	132
8.2	Evaluation results of proposed techniques and baselines according to different cutoffs when query expansion is employed. A bold number indicates the best performance for each cutoff.	132
9.1	Definitions of text containment terms	136
9.2	Text Reuse Categories.	136
9.3	Examples of robustness of DCT fingerprinting	141
9.4	Text reuse detection results in TREC Blogs06 collection. ‘#Sibling’ represents the average number of documents which are related to the detected document through a category.	142
9.5	Text reuse patterns in the TREC Blogs06 collection.	142
9.6	Near-duplicate detection results for two time spans. “ND-detected” denotes tweets involved in at least a near-duplicate relation. “# ND-detected by the same users” means the number of tweets posted by the same users among “ND-detected”. “# ND per ND-detected” denotes the average number of its near-duplicate tweets per ND-detected tweet. “Time gap per ND-pair” denotes the average time difference between posting times of tweets which make a near-duplicate pair.	146
9.7	Examples for near-duplicate types	147
9.8	Results of manual classification	147
A.1	Pseudo-relevance feedback results of the more accurately estimated Fréchet sample mean in the Riemannian manifold defined by the Fisher information metric. GRM ⁺ denotes the pseudo-relevance feedback technique using the more accurately estimated Fréchet sample mean. The results by RM and GRM are borrowed from Table 3.3.	163

B.1	Average query lengths of split topic sets and four Dirichlet smoothing parameters. μ_{short} and μ_{long} are parameters trained for short queries and long queries, respectively. μ_{avgdl} is the average document length. μ_{est} is estimated by our proposed method.	169
B.2	Retrieval results for short queries and long queries according to different Dirichlet smoothing parameters. A number is a MAP score.	170

LIST OF FIGURES

Figure	Page
3.1 Assuming the Euclidean metric space, a $n + 1$ dimensional multinomial distribution is mapped to a point in the n -simplex in Euclidean space (a). Assuming the Riemannian manifold defined by the Fisher information metric, the same point is mapped to a point in the positive n -sphere of radius 2 (b).	25
3.2 Geometric selection algorithm for representing multiple documents in the Riemannian manifold based on the Fisher information metric	30
4.1 The distribution of the number of postings in the blog sites returned by each blog site representation technique	52
4.2 MAP scores for each run of pseudo-cluster selection with a penalty factor by random postings. GR and PCS stand for global representation and pseudo-cluster selection, respectively.	57
5.1 Example of a thread structure	64
5.2 Example of the threaded-view. An indentation indicates a reply relation.	65
5.3 Algorithm for finding all reply relations in a thread. P is a list of postings in chronological order. A is a list of the indices of corresponding parents of postings.	67
5.4 Histogram of normalized location indices	70
5.5 Learning curve on the W3C collection. The change of accuracy on test sets (y -axis) depending on the number of threads in the training set (x -axis) is plotted.	83
5.6 Contexts in a thread structure	84

6.1	Graphs by different construction methods. A circle is a candidate node and a square is a posting node. A number in each square is the identification number of a thread to which the posting belongs.	99
6.2	Two components of new random jump matrix for integrating hierarchical structures into the PageRank algorithm. These two matrices are linearly combined by β , i.e., $\mathbf{E} = (1 - \beta)\mathbf{E}_1 + \beta\mathbf{E}_2$. The red cells indicate random jumps among candidates while the blue cells indicate random jumps among documents (postings). The green cells indicates random jumps within a thread.	102
7.1	In the bottom planes, a small square, a circle and a triangle represent a posting, dialogue context and an author model, respectively. Large shapes denote their geometric mean representations. By Equation (7.8), we find a mean representation of these mean representations as shown in the upper plane.	112
7.2	Illustration of 7.1 in a single plane.	112
8.1	Example of a thread structure. An arrow represents a reply relation.	123
8.2	Greedy Algorithm. S is a posting set to be return to users. k is the maximum size of S . L is a posting list sorted in descending order of posting-level relevance. $route(p_1, p_2)$ is a set of all postings on the route connecting p_1 and p_2	124
8.3	Relevant substructures [Thread ID = 7333]. The redder a node, the more relevant it is. A number in each node is the posting's chronological order.	128
8.4	Relevant substructures [Thread ID = 44226]	129
9.1	DCT fingerprinting	140
9.2	A format of 32bit DCT fingerprint	140
9.3	Distribution of time gaps of near-duplicate pairs in WEEK-SET. (x -axis: time gap (in minutes), y -axis: frequency)	146
A.1	Geometric visualization of the top 20 documents for Topic 770 (GOV2), the arithmetic mean (AM) and the normalized geometric mean (GM) for different metrics, i.e. the Euclidean metric (a) and the Fisher information metric (b).	161

A.2	Determination of a middle point \mathbf{m} on a geodesic linking \mathbf{x} and \mathbf{y}	162
A.3	Relative locations of the more accurately estimated Fréchet sample means. The x -axis corresponds to the relative locations, and the y -axis corresponds to queries for each collection. As a relative location is closer to 1.0, the estimated mean for the topic is located near the normalized geometric mean.	163
B.1	Estimated Dirichlet smoothing parameters (y -axis) according to the numbers of sample terms (x -axis) on the AP collection.	170

CHAPTER 1

INTRODUCTION

Communication via social applications on the Web has emerged as a pervasive social phenomenon. Increasingly popular social applications raise a number of interesting research issues such as how information is propagated over social networks and how social structures in social applications reflect real relationships. Accordingly, social media can be considered from several different research perspectives. For example, in sociology, researchers focus on social dynamics such as how social networks are established and evolved, how social networks can be analyzed, or how online social relationships are related to offline social relationships.

In this work, we view social applications as information sources which can be used to satisfy information needs. Thus, we will discuss why information in social media is valuable and then explore how we can identify relevant information in social media.

What makes information in social media on the Web valuable and unique? Social applications inherit some desirable properties from traditional social media. For example, in social applications, information and knowledge accrue via interactions among members of communities. This process resembles peer-reviewing processes and tends to make information in social applications more reliable. Furthermore, in many social applications, people form an online community by sharing with others who have similar interests. Since a small number of topics are typically discussed in depth, such online communities or social applications can be considered useful information sources for these topics. In addition, social applications often carry unfiltered opinions. For example, in social applications such as Twitter and blogs, people

tend to express themselves freely and create postings carrying frank opinions about subjects that we rarely hear about from public media. These opinions can help us understand a complex topic.

Another advantage of information in social applications comes from its accessibility. In contrast to off-line meetings or conversations, communication in most social applications is non-volatile, with the records being easily accessible even after the discourses are finished. Furthermore, information in most Web-based social applications, except for a few private applications such as chat and email, are publicly accessible. Consequently, social applications can be considered to be publicly available information resources.

Even one of the many existing social applications can provide abundant information. To effectively leverage social applications as information sources, efficient tools that can identify relevant information are necessary, i.e., a good search engine. However, search algorithms used for general web pages often overlook unique features of social applications which may prove helpful for search. That may be why the search quality for social media is not as good as that for general web pages in many web search engines. Therefore, we propose to investigate advanced search algorithms that use unique features and structures for each social application.

1.1 Structures in Social Media

The quality of the information and its accessibility make social media valuable. Accordingly, social applications are often designed to systemically support these properties. That is, each social application is designed to effectively deliver opinions to other people, to encourage people to participate in discussions, and to help people access information. In many cases, these intentions are achieved via explicit or implicit structures in the social applications. Of the many structures embedded in social ap-

plications, there are three that are both common across applications and important. These are social structures, hierarchical structures, and conversational structures.

Social structures represent social relationships between community members. For example, in online forums, a useful criterion provided by a social structure is whether or not a member is an expert in a specific topic. Many roles such as friends in Facebook, followers in Twitter, and blog rolls in blogs also define interesting social relationships. Hierarchical structures correspond to the way that information is organized. For example, a blog consists of categories and postings. An online forum contains many subforums that have many threads, which in turn consist of postings. Conversational structures are formed by conversation-like behaviors for discussion and feedback in social applications. For example, relations formed by replies in blogs, forums, emails and Facebook establish discourses. Community-based question answering (CQA) services have conversational structures via questions and answers.

Taking account of these structures, we can identify unique characteristics of various types of social applications as follows:

- **Forum**

A forum is a community where people who are interested in a specific topic gather and have discussions. Therefore, intrinsically, a forum can be considered as a topic-centric document set. A boundary of a community is definite, i.e. separated by members or non-members. Some forums are public while others are exclusive. The latter cases tend to have stronger participant boundaries. Regardless of the strength of the boundary, social structures on a forum can be usually well defined. Most forums have hierarchical structures. A forum has many sub-forums according to broad topic categories. A sub-forum has many threads. A thread can be considered a minimal topical unit to address a specific topic. People who are interested in the topic reply to the preceding postings in the thread. These reply relations establish a conversational structure

in a thread. Therefore, forums usually have both hierarchical structures and conversational structures.

- **Blog**

A blog is a publishing application which is owned and operated by a few people, i.e. bloggers. Blogs are usually topic-centric in that they address a small number of topics. While identities of writers are known, readers can be anonymous because any one can read postings by subscribing to feeds. Thus, social structures are vague. However, we can analyze social relationships with some degree of limitation by looking into links between bloggers such as blog rolls. Blogs have hierarchical structures according to categories defined by the blogs' owners. On the other hand, other structures are not distinct. We can sometimes see that replies to postings have conversational structures. However, the replies are usually short and the conversational structures are not necessarily expected to exist in contrast to forums.

- **Community-based Question Answer (CQA)**

A community-based question answer (CQA) service is a special type of forum that focuses on question-answer interactions. CQA services are usually operated by commercial search portals (e.g., Yahoo! Answers) and many users can ask questions or post answers to the questions because most CQA services are public. Although the community boundary is not obvious, we can find some social structures because identification information such as user ID's is known. Furthermore, since most CQA services are not limited to specific topics, the services provide hierarchical structures according to well-defined categories to organize many topics. In CQA threads, there is usually a simple flat conversational structure, that is, one person posts a question while others answer

the question. Although some CQA services also support discussions by replies, conversations in most CQAs happen as question-answer pairs in flat structures.

- **Emails and Chat**

Emails and Chats are private social applications. The community boundary is small and can be easily determined. Although emails may be organized according to categories defined by owners, hierarchical structures do not really exist. On the other hand, they naturally have conversational structures through replies. Chats almost always are volatile, and furthermore, emails and chats can only be accessed by the direct participants in most cases. Therefore, using these information sources is limited to a few private applications such as desktop search or personal information management.

- **Microblogs**

Microblogs, e.g., Twitter¹ are a special form of blogs, and has recently become one of the most popular online social networking tools because of the convenience of usage. Microblogs have some interesting aspects that differ from general web pages or blogs. First, since only short text is allowed to be able to be easily typed even by mobile devices and messages are delivered to followers with little latency, each message tends to be “instant”. That is, many people use microblogs to express their immediate reactions and opinions, and report facts rather than to record persistent information that typically involves more formal writing. This property leads to considering tweets as interesting resources for detecting temporal or emerging issues. Second, social structures are more definite. While feed subscribers in blogs are not known, followers in Twitter are known. This can define social relationships between readers and authors. Third, Twitter does not have rigid hierarchical structures. Nevertheless,

¹<http://twitter.com/>

since tagging is very popular, a tag can give a hint about categories that tweets may be associated with. Fourth, conversational structures are supported by a unique mechanism called “mention” and “reply”. A tweet containing special tags for this mechanism can be considered an explicit utterance.

- **Hybrid Applications**

Some social applications such as Facebook² contain various types of social applications mentioned above.

1.2 Major Retrieval Challenges

All of three social media structures can help us not only to better understand social applications but also to improve retrieval performance. Hierarchical structures can be used to represent a collection of individual information units, social structures can be used to identify characteristics of community members, and conversational structures can be used to clarify the purpose of discourses and information. However, these structures are not always explicit. Hierarchical structures are often explicit because they are usually defined by layouts of HTML pages or special tags. On the other hand, social structures and conversational structures are sometimes implicit. For example, a social network associated with a blog is somewhat vague in contrast to Facebook’s network. Even when an explicit social network is recognized, sufficient information for identifying the characteristics of participants may not be given because many applications assume anonymity of participants. In the case of conversational structures, even online forums where such structures in threads are important often collapse the structures and display postings just in chronological order. Accordingly, we need to discover these useful structures for given social applications before performing retrieval.

²<http://www.facebook.com/>

Once these useful structures are discovered, relevant contexts should be extracted from the structures because not all the information embedded in these structures are relevant. For example, in blog site search that we will discuss in Chapter 4, we can easily discover a hierarchical structure by relations between a blog and its member postings. However, we do not need to consider all the postings to find relevant blog sites because even a single blog site addresses various topics. Therefore, we need to extract or consider only relevant postings considering the hierarchical structure. Similarly, in a forum thread containing a conversational structure, not all the conversations in the thread are relevant; thus, relevant conversations or their parts need to be extracted. In addition, for more precise representations of contexts, we sometimes need to control the granularity of contexts. For example, if a context is too coarse, it may be too noisy. On the other hand, if a context is fine-grained, it may not be capable of capturing relevant information sufficiently.

A retrieval object and its various contexts should be represented in appropriate ways so that they can be exploited in retrieval frameworks. For example, a blog site can be represented by a coarse-grained context. On the other hand, we can make a representation using a number of fine-grained contexts. Also, contexts extracted from different structures can be used to make a representation. Therefore, we need to develop an effective framework to address these various representations.

These major retrieval challenges in search using social media structures can be summarized as follows:

- Discovery of social media structures
- Extraction of relevant contexts from social media structures
- Representations for multiple relevant contexts

Considering these challenges, we first theoretically justify a framework to represent multiple contexts using the geometric mean. This framework is used throughout this

work for different contexts extracted from social media structures. Then, for each structure, we discuss how to discover the structure in social applications and how to extract relevant contexts from the discovered structures. Using the framework for multiple contexts representation, we present retrieval algorithms that exploit the structures or the contexts. To evaluate these techniques, we consider various tasks for social applications. For each task, we obtain data from real applications and discuss how to build test collections with queries and relevance judgments.

1.3 Minor Challenges

Besides the major retrieval challenges, there are many more interesting and important challenges related to social media search. Among them, we address two additional challenges in this thesis.

The ultimate goal of the major retrieval challenges is to find a retrieval object including relevant information. For example, the object can be a blog site or a forum thread. However, even if we can locate a relevant retrieval object, it may not be easy to find relevant information in the object. This is especially true for set objects which consist of multiple small objects, e.g., a thread consisting of postings. These set objects are often so large that users spend too much time finding relevant information by reading all the contents. Therefore, in order to satisfy users' information needs more quickly, we need to address identification of relevant substructures in large set objects.

In addition to the three structures mentioned previously, there are many other social media structures having the potential to be exploited for various retrieval tasks. For example, text reuse structures in social applications can help retrieval in direct or indirect ways. Users in web applications including social applications often borrow text from other sources. We call these actions or the results "text reuse". Text reuse can happen in many different ways, e.g., by putting an excerpt from a news article

in a posting or illegally copying text. Note that RT or re-tweet in microblogs is a mechanism that allows users to legally take these actions. Also, users sometimes intentionally spread some specific messages such as spam over many social applications. We can infer interesting relations among documents sharing common text. We call a structure constructed by these relations a text reuse structure. By looking into these text reuse structures and patterns of text reuse in social applications, we can understand social applications better and get insights for better retrieval algorithms. For example, we try to detect the original source of reused text by tracing the information flow appearing in a text reuse structure. If it comes from a document of a specific user in the same social application, this fact can be a signal that the user is an authoritative user. Also, when delivering search results, we can present only the original document. Moreover, we can use reused text appearing across multiple social applications for inferring links of users, contents and topics among the applications. Addressing all the applications of text reuse structures is beyond the scope of this thesis. Therefore, we discuss how to detect text reuse and analyze text reuse patterns in real social applications. We expect our work to inspire future research focusing on text reuse structures in social applications.

The two minor retrieval challenges that we address in thesis are summarized as follows:

- Identification of relevant substructures in set retrieval objects
- Discovery of text reuse structures and text reuse pattern analysis

1.4 Contributions

Our major contributions in this work are as follows:

- An understanding of unique structures in social media applications which imply social information and community knowledge

- Algorithms for discovering explicit or implicit structures in social media applications and extracting useful contexts from the structures
- A geometry-based representation model for multiple contexts
- Retrieval models incorporating information extracted from social media structures to improve the effectiveness of search
- Evidence showing that social media structures can be helpful resources for utilizing social applications as information sources
- Customization of retrieval models for various real-world applications
- Practices for building test collections for social media search evaluation

1.5 Organization

In Chapter 2, we review previous work on social media search, including research related to each unique social media structure. Chapter 3 to 7 address our major retrieval challenges. In Chapter 3, we theoretically justify a geometry-based representation framework. This framework is used for representing multiple contexts extracted from various structures through this thesis. Chapter 4, 5, and 6 address how to define and exploit hierarchical, conversational and social structures, respectively. Specifically, each structure is paired up with a real task where the structure plays an important role. That is, hierarchical structures are addressed via blog site search. On the other hand, conversational structures and social structures are addressed via forum search and expert finding, respectively. In Chapter 7, we propose a technique combining all the three structures. Chapter 8 and 9 describe the research related to the minor challenges. In Chapter 8, we focus on relevant substructures embedded in retrieval objects of social media search. In Chapter 9, text reuse structures are introduced and text reuse patterns in blogs and microblogs are analyzed. Finally, in

chapter 10, we conclude this thesis with a brief summary and a discussion of future research directions.

CHAPTER 2

RELATED WORK

There has been relatively little work to date exploring the use of social media structures to improve search in social media. For some of the “older” social media and specific structures in those media, such as the sender/receiver structure in emails, there has been some prior work germane to this proposal. For example, the release of email collections such as the Enron email corpus [64] and the TREC W3C email corpus [124] has encouraged many researchers to study certain aspects of emails including conversational structures captured in threads [103, 134, 138]. As public social applications such as blogs and forums flourish, studies about these applications are forming a growing stream of social media research. Specifically, public blog test data released by the Text REtrieval Conference (TREC) has initiated research on IR for blogs [83]. In addition, Twitter is attracting many researchers because of its unique real-time characteristics. In the remainder of this chapter, we will review related work addressing the structures of social media. Note that other references specific to each topic are described in the relevant chapters, but this chapter points out significant work in the general area of social media structures. Also, in the last section, we list our own published work related to this thesis.

2.1 Hierarchical Structure

Objects used in search applications often possess a natural hierarchical structure. For example, even a short document comprises a number of sentences. Accordingly, exploiting hierarchical structures has been frequently addressed in IR.

One way to consider hierarchical structures is to combine fine-grained multiple evidence to represent collective evidence. For example, various combination heuristics suggested by Fox and Shaw [42] and analyzed by Lee [73] continue to be used in many IR tasks such as passage retrieval and resource selection. Also, in distributed Information Retrieval, resource selection techniques combine multiple documents to represent a collection [135, 19, 120]. Using passage-level evidence [18, 78, 8] for document retrieval necessarily employs combination techniques for hierarchical structures. Some approaches leverage clustering techniques for constructing hierarchical structures. Xu and Croft [136] demonstrated that topic-based retrieval using clustering is effective for resource selection. Liu and Croft [80, 81] introduced cluster-based language model representation techniques. Recently, Seo and Croft [114] analyzed representations for multiple documents via Information Geometry and proved that a combination technique of hierarchical evidence by the geometric mean can be superior to other combination techniques.

Another way to take advantage of hierarchical structures is to integrate global contexts into representations for fine-grained objects. For example, we can employ a multi-stage smoothing technique [140] to integrate a document model with a cluster or a collection to which the document belongs. While Liu and Croft [79] proposed a document model integrated with a cluster, Ogilvie and Callan [93] introduced a hierarchical entity model for XML retrieval. In addition, the INEX (Initiative for the Evaluation of XML retrieval) Ad Hoc Tack focused on hierarchical structures provided by XML markups for finding relevant information [45].

In the social media search literature, there are some recent studies addressing hierarchical structures. Arguello et al. [4], Elsas et al. [33] and Seo and Croft [112, 110] introduced various blog representations combining postings or feeds in each blog. Also, Elsas and Carbonell [34] and Seo and Croft [115] showed that a thread in online forums can be effectively represented by its postings.

2.2 Conversational Structure

We can identify conversational structures explicitly or implicitly in most social applications involving interactions between users. For example, in emails and forums, conversational structures are formed by replies. However, the fact that the structures in many applications are often collapsed creates a challenge in leveraging them. To tackle this problem, there have been efforts known as thread structure discovery or disentanglement. Lewis and Knowles [75] are among the first who have focused on threading email conversations. Smith et al. [121] proposed a new application design to implement threaded chats. Yeh and Harnly [138] and Erera and Carmel [36] discussed similarity matching techniques for email thread detection. There are similar attempts in domains other than emails. Elsner and Charniak [35] and Wang and Oard [130] studied conversation disentanglement in online chat dialogues. Wang et al. [131] pursued thread structure discovery in newsgroup style conversations. Recently, Cong et al. [76] modeled semantics and structures of threads by minimizing a loss function based on assumptions for sparsity of topics and reply relations.

Some researchers have focused on finer-grained discourse acts rather than simple reply-based thread structures. Shrestha and McKeown [119] introduced techniques for identifying question-answer (QA) pairs in an email conversation for email summarization. Cong et al. [26] also investigated finding QA pairs in online forums. One of the purposes of these attempts is to augment CQA archives. While the amount of data for CQA is limited, there are plenty of forums that can be rich information sources. If we can systemically extract QA pairs from forums, then we can significantly expand the coverage of CQA. In contrast, Carvalho and Cohen [21] focused on more general acts in emails such as request, propose, data, and so on.

There have been efforts for leveraging conversational structure for retrieval. For example, the University of Maryland group [134, 77, 92] tried to use simple thread information for email distillation tasks in the TREC enterprise track. Wanas et al.

[129] studied quality-based rankings using a couple of thread-based features. Seo and Croft [115] extracted various contexts from forum threads and exploited them for thread search tasks as well as posting search tasks.

2.3 Social Structure

A social structure is one of the most crucial features distinguishing social applications from general Web applications. Since social structures reflect the relationships among people in communities, these structures can provide richer contexts that we cannot otherwise easily obtain from text such as a posting or a thread. For example, social roles [47] of members in a community can be identified by observing social structures. Fisher et al. [39] and Welser et al. [133] analyzed and visualized social roles in online communities such as Usenet newsgroups. In particular, they defined several distinguishing social roles; e.g., answer person, question person or discussion person. Gleave et al. [47] introduced strategies for identifying social roles in online communities by extending the previous studies. Welser et al. [132] applied these approaches to community Q&A systems to identify “expert” roles. In addition, Viégas [128] focused on visualizing social structures including social roles in online social archives. McCallum et al. [86] proposed a generative model to capture latent author roles as well as topics in email archives.

Among the many social roles in online communities, many expert roles assume particular importance when we view an online community as an information source, because opinions of experts can be considered to be more reliable and informative than those of newbies in the community. Accordingly, there have been abundant studies of expert identification. In many general Web studies, the PageRank [95] and HITS [63] algorithms are among the most frequently referenced techniques. These general graph-based algorithms for finding authoritative sources via hyperlink structures on the Web can be applied to social media applications. Campbell et al. [20] employed

graph-based ranking algorithms to identify experts in an email network. Zhang et al. [141] reviewed expertise ranking algorithms and performed modeling of social network in an online forum using simulation techniques. Jurczyk and Agichtein [58] used a link analysis algorithm to rank authors in community based-QA portals. Seo and Croft [113] showed that link analysis on a graph modeling thread structures and social structures can be a promising approach for finding experts in online forums. Fu et al. [44] introduced an expertise propagation algorithm for an email network. Lappas et al.'s work [70] addressed team formation problems while considering the expertise of individuals in a social network.

Since 2005 the TREC community has organized an expert finding task in a virtual enterprise environment [124]. This task employed an email archive. According to reported results for the TREC expert finding task, link-based techniques were not as effective as language modeling-based techniques for their collection. For example, Balog et al. [6] detailed a language modeling framework for expert finding. In addition, Serdyukov et al. [116] introduced relevance propagation modeling through author nodes and document nodes for this task.

Aardvark [54] is a successful social application employing expert finding techniques. When a user posts a question, the Aardvark search engine locates relevant people who may answer the question taking into consideration social relationships as well as user profiles.

2.4 Our Published Work

Much of the research presented in this thesis has been published in the following references.

- Seo, Jangwon, and Croft, W. Bruce. Geometric Representations for Multiple Documents. In the Proceedings of the 33rd Annual International ACM SI-

GIR Conference on Research and Development in Information Retrieval (SIGIR 2010), pp. 251-258, 2010.

- Seo, Jangwon, and Croft, W. Bruce. Unsupervised Estimation of Dirichlet Smoothing Parameters. In the Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010), pp. 759-760, 2010.
- Seo, Jangwon, Croft, W. Bruce, and Smith, David A. Online Community Search Using Thread Structure. In the Proceedings of the ACM Conference on Information and Knowledge Management (CIKM 2009), pp. 1907-1910, 2009.
- Seo, Jangwon, and Croft, W. Bruce. Thread-based Expert Finding. In the SIGIR 2009 Workshop on Search in Social Media (SSM 2009), 2009.
- Seo, Jangwon, and Croft, W. Bruce. UMass at TREC 2008 Blog Distillation Task. In the online Proceedings of the 2008 Text REtrieval Conference (TREC 2008), 2009.
- Seo, Jangwon, and Croft, W. Bruce. Blog Site Search Using Resource Selection. In the Proceedings of the ACM Conference on Information and Knowledge Management (CIKM 2008), pp. 1053-1062, 2008.
- Seo, Jangwon, and Croft, W. Bruce. Local Text Reuse Detection. In the Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008), pp. 571-578, 2008.
- Seo, Jangwon, and Croft, W. Bruce. UMass at TREC 2007 Blog Distillation Task. In the online Proceedings of the 2007 Text REtrieval Conference (TREC 2007), 2008.

CHAPTER 3

GEOMETRIC REPRESENTATIONS FOR MULTIPLE CONTEXTS

Social media applications contain various explicit or implicit structures. In this thesis, we often represent retrieval objects in a social application by combining multiple contexts extracted from these different structures. In fact, making representations using multiple contexts or documents is a typical approach in Information Retrieval (IR). For example, tasks such as relevance feedback, passage retrieval and resource selection in distributed information retrieval or in aggregated search, use representations for sets of multiple documents.

One standard approach for relevance feedback is to estimate an underlying relevance model from given feedback documents and sample likely terms from the model for query expansion. That is, the estimated underlying model can be considered as a representation of the feedback documents. In passage retrieval, representations of text passages can be used to rank passages or documents. In the latter case, we represent a document using a combination of some or all of its passages. In resource selection tasks, the resource or collection is represented using the documents in the collection.

As many tasks require representations for multiple documents, various approaches have been introduced. Among them, representation techniques based on the arithmetic mean and concatenation are frequently used. Representation techniques based on the arithmetic mean literally compute the arithmetic mean of multiple language models or vector representations. For example, the Rocchio algorithm for relevance

feedback [106] combines feedback document vectors by the arithmetic mean. Representation techniques based on concatenation make a large document by concatenating multiple documents and use a language model or vector to represent the large document. For example, the large document model by Arguello et al. [4] represents a blog by concatenating all feeds in the blog.

In addition to traditional group representation techniques, some recent studies show the potential of a new representation technique, the geometric mean representation of language models [81, 34]. Liu and Croft [81] compared representation techniques for cluster retrieval and demonstrated that representations using the geometric mean outperformed others via empirical evaluation. Kogan et al. [65] used the geometric mean for k -means clustering.

The previous work which uses the geometric mean to represent a group of documents, however, did not theoretically analyze the geometric mean in the language modeling framework. In other words, although they have demonstrated the performance of representation techniques based on the geometric mean empirically, theoretical evidence or the assumptions behind the geometric mean have not been sufficiently addressed to understand its value in IR.

We also, in this thesis, use geometric mean-based representations because they have often produced better retrieval performances for various tasks that we will address. However, using these representation techniques without any theoretical justification can lead to the misuse of the techniques. Therefore, in this chapter, we give a theoretically grounded explanation for geometric mean-based techniques for representing multiple documents objects which can be expressed as multinomial distributions. To do this, we consider Information Geometry as a tool and discuss how the arithmetic mean as well as the geometric mean can be interpreted in certain geometries. More specifically, we show that both the arithmetic mean and the geometric mean that are prevalently used for multiple document representations in IR relate to

the Fréchet sample mean which minimizes the Fréchet sample function. Indeed, the Fréchet sample mean is a general definition for a point representing multiple points in a metric space. Therefore, we can observe which metric space produces empirically the most effective representation, by considering different metric spaces. As a result, we show that the geometric mean is closer to the Fréchet mean in the Riemannian manifold defined by the Fisher information metric.

In addition, we address two generic IR applications considering the geometric interpretation: cluster retrieval and pseudo-relevance feedback. Particularly, for pseudo-relevance feedback, we introduce a variation of the relevance model [71], the geometric relevance model, and show that this new approach performs better than the relevance model.

Based on these results, we will leverage the geometric mean-based techniques as a framework for combining multiple structural contexts in the next chapters. In fact, since these contexts can have a form of a pseudo-document or a multinomial distribution, we can apply our proposed representation technique to our social media search tasks without loss of generality.

3.1 Related Work

Combining multiple evidence is one of the most frequently addressed topics in Information Retrieval. Belkin et al. [7] showed that different representations of the same information object leads to different results and combinations of such representations can improve retrieval performance. Various combination heuristics suggested by Fox and Shaw [42] and analyzed by Lee [73] are still used in many IR tasks such as passage retrieval and resource selection. Using passage-level evidence [18, 78, 8] for document retrieval necessarily requires combination techniques. Resource selection where a collection is represented by its own documents [19, 120] actively uses combination techniques as well.

Relevance feedback (and pseudo-relevance feedback) is another task using combination-based representation techniques. To estimate a query model for query expansion, the top ranked documents are combined. Rocchio [106] introduced a feedback technique to combine positive or negative feedback documents in vector spaces. Lavrenko and Croft [71] introduced a technique that estimates an underlying relevance model in the language modeling framework. In fact, these standard relevance feedback approaches implicitly use the arithmetic mean. Recently, Collins-Thompson and Callan [25] used a parametric approach using re-sampling to estimate a posterior Dirichlet distribution for the documents. That is, they use the mean and the variance of the Dirichlet distribution to get a feedback model.

The geometric mean-based representation technique was relatively recently introduced. Liu and Croft [81] demonstrated that representation by the geometric mean works well for cluster retrieval via comparisons with various representation techniques. The geometric mean is often used in other fields. For example, Kogan et al. [65] used the geometric mean for k -means clustering. Veldhuis [127] showed that a centroid of the symmetrical Kullback-Leibler divergence is related to the arithmetic mean and the normalized geometric mean.

In this chapter, to justify the use of the geometric mean in IR, we find evidence from Information Geometry. Rao [104] and Jeffreys [56] are the first people who considered the Fisher information metric as a Riemannian metric. Later, Efron [30] focused on differential geometry in statistics considering the curvature of statistical models. Recently, Lebanon [72] applied the theory to many machine learning tasks. See Amari and Nagaoka [3] and Kass and Vos [60] for comprehensive introduction to Information Geometry.

3.2 The log-linearity of the geometric mean

Before discussing theoretical justifications about the geometric mean, we begin with an intuitive explanation why the geometric mean should have advantages for many IR tasks. The most critical reason that the geometric mean works is its log-linearity. As more documents contain a specific term, the geometric mean for the term increases exponentially while the arithmetic mean increases linearly.

For example, assume that tf 's are similar for terms and 5 documents are given. If term A is contained in only one document and term B is contained in two documents, the difference between the geometric means associated with the terms is small. However, if term C is contained in 4 documents and term D is contained in 5 documents, the difference between the geometric means is large. In both cases, the differences between the arithmetic means are uniform.

Accordingly, the arithmetic mean can be sensitive to a few dominant terms in a small number of documents. On the other hand, the geometric mean favors the common terms across a whole set of documents and is relatively insensitive to a few dominant terms. This property has been shown empirically to be desirable for multiple IR applications [81, 34].

3.3 Geometry of Multiple Documents¹

We introduce the Fréchet mean which is a generalized mean that can be defined in any metric space because we want to consider different metric spaces. For example, the Fréchet mean in the Euclidean metric space is the ordinary mean that we usually use. In this chapter, to see how can the arithmetic mean and the geometric mean can

¹This geometry is applicable to any contexts which can be represented by language models or multinomial distributions as well as documents. However, just for convenience, we use term “document” in this chapter.

be derived, we consider two different metric spaces, i.e., the Euclidean metric space and the Riemannian manifold defined by the Fisher information metric.

3.3.1 Fréchet Mean

Let us consider a Riemannian manifold \mathcal{M} with a distance measure $dist(\mathbf{x}, \mathbf{y})$ where \mathbf{x} and \mathbf{y} are points on the manifold. Assume that we have a distribution Q on a convex set $\mathcal{U} \subset \mathcal{M}$. Now we define a function $F : \mathcal{M} \rightarrow \mathbb{R}$ as follows:

$$\Phi(\mathbf{c}) = \int_{\mathbf{p} \in \mathcal{U}} dist^2(\mathbf{c}, \mathbf{p})Q(d\mathbf{p}) \quad (3.1)$$

where \mathbf{c} is a point in \mathcal{M} .

This function is known as the Fréchet function. A set of points which minimize the function is called the Fréchet mean set of Q . If there is only a point in the set, the point is called the Fréchet mean. This general notation for a center or centroid associated with a probability distribution was introduced by Fréchet [43] and Karcher [59]. This mean is called by various names, e.g., the center of mass, barycenter, Karcher mean and Fréchet mean. In this work, we refer to this mean as the Fréchet mean². The concept of the Fréchet mean is general and not limited to any specific metric; accordingly, this can be applied to any metric space. Indeed, as we will see soon, it also generalizes the ordinary Euclidean mean.

Kendall [62] proved that if the support of Q is in a geodesic ball of sufficiently small radius r , then one Fréchet mean uniquely exists. As we see later, we consider a statistical manifold for multinomial distributions, and the distributions are mapped onto a simplex or a positive sphere. Since the mapped area is sufficiently small, a unique Fréchet mean exists. For example, in case of a sphere, the radius of the geodesic ball is $\pi/4$ and the positive sphere is contained in the ball.

²Strictly speaking, this is the intrinsic Fréchet mean in that we use a geodesic distance. However, since we address only the intrinsic Fréchet means in this work, we omit term “intrinsic”.

If we have n unique points $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ in m i.i.d. samples from distribution Q , then we consider the sample Fréchet mean which minimizes the Fréchet sample function given by

$$\bar{\Phi}(\mathbf{c}) = \sum_{i=1}^n \text{dist}^2(\mathbf{c}, \mathbf{p}_i) \hat{Q}(\mathbf{p}_i) \quad (3.2)$$

where \hat{Q} is an empirical distribution estimated from the samples.

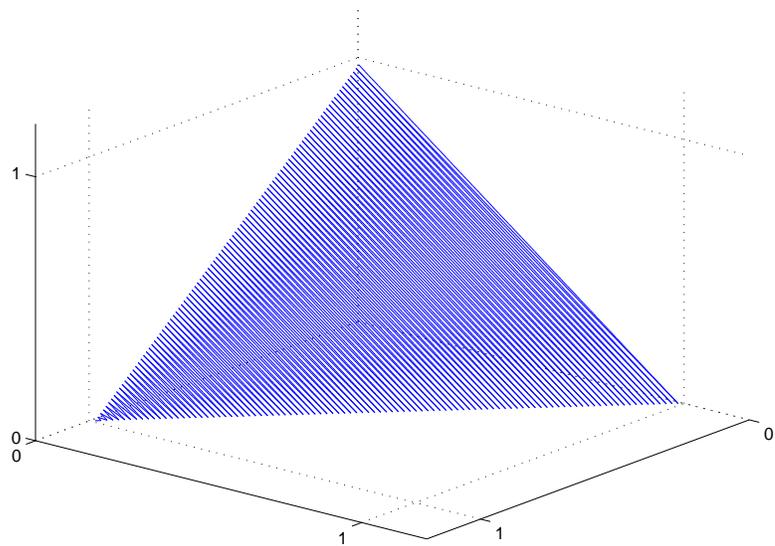
Bhattacharya and Patrangenaru [11] showed that every measurable choice from the Fréchet sample mean set of \hat{Q} is a strongly consistent estimator of the Fréchet mean of Q . In this chapter, we consider multiple documents to represent as samples and the Fréchet sample mean as a representation. Therefore, we address how to compute the sample Fréchet mean from the multiple documents in the following sections.

3.3.2 Euclidean Metric space

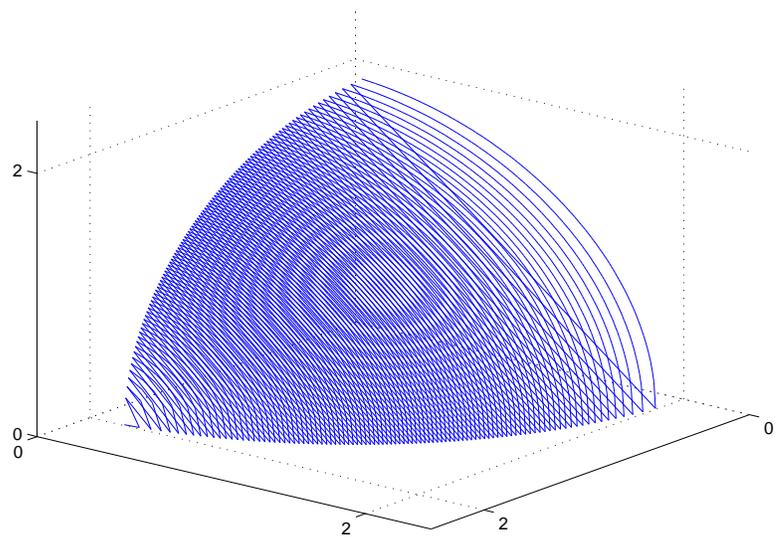
Let's begin with the Euclidean metric space. We assume that terms observed in a document are samples from a multinomial distribution and each document has a distinct distribution. Assuming a conjugate Dirichlet prior, we estimate the multinomial distribution, i.e. a language model, using Dirichlet smoothing [139] as follows:

$$\text{Pr}(w|D) = \frac{tf_{w,D} + \mu \cdot cf_w / |C|}{|D| + \mu} \quad (3.3)$$

where $tf_{w,D}$ is the occurrence of term w in document D , cf_w is the occurrence of w in a set of observations C considered for the prior distribution (typically, a corpus), $|D|$ is the number of observations, i.e. the length of D , $|C|$ is the length of C , and μ is the Dirichlet smoothing parameter. Note that $Pr(w|D)$ is a parameter which corresponds to outcome w in the multinomial distribution.



(a)



(b)

Figure 3.1. Assuming the Euclidean metric space, a $n + 1$ dimensional multinomial distribution is mapped to a point in the n -simplex in Euclidean space (a). Assuming the Riemannian manifold defined by the Fisher information metric, the same point is mapped to a point in the positive n -sphere of radius 2 (b).

The size of vocabulary of a language model is defined as the number of terms observed in C , which also determines the number of dimensions of the Euclidean metric space for a multinomial distributions. When the number of dimensions is $n + 1$, a multinomial distribution corresponds to a point in n -simplex \mathcal{P}_n which is defined as follows:

$$\mathcal{P}_n = \left\{ \mathbf{x} \in \mathbb{R}^{n+1} : \forall i, x^{(i)} > 0, \sum_{i=1}^{n+1} x^{(i)} = 1 \right\} \quad (3.4)$$

An example of 2-simplex embedded in 3-dimensional Euclidean space is shown in Figure 3.1.

Since a geodesic linking two points in n -simplex is a straight line, the distance between two multinomial distributions is calculated by the Euclidean distance as follows:

$$dist(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n+1} (x^{(i)} - y^{(i)})^2} \quad (3.5)$$

Consider multinomial distributions of k given documents, $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k$ as samples from distribution Q over the n -simplex. Then, the Fréchet sample function is given by

$$\bar{\Phi}(\mathbf{c}) = \sum_{i=1}^k \hat{Q}(\mathbf{p}_i) \sum_{j=1}^{n+1} (c^{(j)} - p_i^{(j)})^2 \quad (3.6)$$

Therefore, we have the following optimization problem to obtain the Fréchet sample mean.

$$\begin{aligned}
&\text{minimize} && \sum_{i=1}^k \hat{Q}(\mathbf{p}_i) \sum_{j=1}^{n+1} (c^{(j)} - p_i^{(j)})^2 \\
&\text{subject to} && \sum_{j=1}^{n+1} c^{(j)} = 1 \\
&&& c^{(j)} > 0 \quad \forall j
\end{aligned}$$

It is trivial to solve this problem using the method of Lagrange multipliers. Finally, we have a solution as follows:

$$c^{(j)} = \sum_{i=1}^k p_i^{(j)} \hat{Q}(\mathbf{p}_i) \tag{3.7}$$

This is the Fréchet sample mean in the Euclidean metric space. Indeed, if $\hat{Q}(\mathbf{p}_i)$ is uniform, i.e, $1/k$, then this is the same as the ordinary Euclidean mean or the arithmetic mean. Therefore, the Fréchet sample mean in the Euclidean metric space generalizes the arithmetic mean.

We use the Fréchet sample mean as a representative multinomial distribution for the given group of multiple documents.

3.3.3 Riemannian manifold defined by the Fisher information metric

Many IR approaches assume that data is embedded in the Euclidean geometry. However, assumptions of non-Euclidean geometries may lead to a better understanding of data. We here consider a Riemannian space where a Riemannian metric is the Fisher information metric. This metric space is used for investigating the geometric structures of statistical models in most of the Information Geometry literature [104, 3, 60]. Furthermore, a number of approaches assume this metric space for statistical inference and machine learning [68, 72, 3]. Particularly, for text classification, Lafferty and Lebanon [68] showed that techniques based on this metric space perform better than techniques based on the Euclidean metric.

The Fisher information metric is defined as follows:

$$\begin{aligned} g_{i,j}(\boldsymbol{\theta}) &= \int \frac{\partial \log p(x; \boldsymbol{\theta})}{\partial \theta^{(i)}} \frac{\partial \log p(x; \boldsymbol{\theta})}{\partial \theta^{(j)}} p(x; \boldsymbol{\theta}) dx \\ &= E_{\boldsymbol{\theta}} \left[\frac{\partial \log p(x; \boldsymbol{\theta})}{\partial \theta^{(i)}} \frac{\partial \log p(x; \boldsymbol{\theta})}{\partial \theta^{(j)}} \right] \end{aligned}$$

where $\boldsymbol{\theta}$ is a point in a differential manifold and corresponds to a statistical model in a parametric family $p(x; \boldsymbol{\theta})$, i and j are indices for a coordinate system. In this work, it is easy to think that $\boldsymbol{\theta}$ is a multinomial model for a document while i and j are indices for unique terms in vocabulary.

This metric has some nice properties. By Cramér-Rao inequality [104], the variance of unbiased estimators is bounded by the inverse of the metric. Particularly, an unbiased estimator achieving the bound is called an efficient estimator which is the best unbiased estimator because it minimizes the variance. Furthermore, by Chentsov's theorem [23], the Fisher information metric is the only Riemannian metric which is invariant under basic probabilistic transformations.

We now look into the Riemannian geometry with the Fisher information metric as a Riemannian metric. First of all, let us consider the positive n -sphere of radius 2, $\tilde{\mathcal{S}}_n^+$ instead of n -simplex P_n .

$$\tilde{\mathcal{S}}_n^+ = \left\{ \mathbf{x} \in \mathbb{R}^{n+1} : \forall i, x^{(i)} > 0, \sum_{i=1}^{n+1} (x^{(i)})^2 = 2^2 \right\} \quad (3.8)$$

Figure 3.1 shows an example of the positive 2-sphere of radius 2.

We can define transformation $\phi : \mathcal{P}_n \rightarrow \tilde{\mathcal{S}}_n^+$ by

$$z^{(j)} = \phi(\mathbf{x})^{(j)} = 2\sqrt{x^{(j)}} \quad (3.9)$$

The inverse transformation ϕ^{-1} is well known to pull back the Fisher information metric on \mathcal{P}_n to the Euclidean metric on $\tilde{\mathcal{S}}_n^+$ [60, 72]. Therefore, the transformation

is an isometry, and we can compute the distance between two statistical models by the Fisher information metric using the geodesic distance between two corresponding points on the sphere. In other words, the distance is the length of the shortest curve linking two corresponding points on the sphere and is given by

$$\text{dist}(\mathbf{x}, \mathbf{y}) = 2 \arccos \left(\sum_{j=1}^{n+1} \sqrt{x^{(j)}y^{(j)}} \right) \quad (3.10)$$

This is called the information distance.

With this distance, we have the following Fréchet sample function.

$$\bar{\Phi}(\mathbf{c}) = 4 \sum_{i=1}^k \arccos^2 \left(\sum_{j=1}^{n+1} \sqrt{x^{(j)}y^{(j)}} \right) \hat{Q}(\mathbf{p}_i) \quad (3.11)$$

Unfortunately, there is no closed form solution for the Fréchet sample mean which minimizes this function. Although we can use some convex optimization techniques, such approaches may be impractical in case that n is large. Indeed, in many IR tasks, $n + 1$ is the size of vocabulary and can be very large.

Instead, we consider an approximation to the Fréchet sample mean. Via the proof in Appendix A, we can get two approximation points, i.e., the arithmetic mean and the normalized geometric mean. We take the following approach to decide a better representation among them. Figure 3.2 describes the algorithm. This algorithm allows us to choose a point which is closer to the Fréchet sample mean as a representation. We call this approach “geometric selection”. We will see how this approach works for representing multiple documents through experiments.

3.4 Experiments

To evaluate representation techniques derived in the previous section, we conduct experiments for two different tasks: cluster retrieval and pseudo-relevance feedback.

- | |
|--|
| <ol style="list-style-type: none"> 1. Compute the arithmetic mean \mathbf{c}_A and the normalized geometric mean \mathbf{c}_G from multinomial models of multiple documents. 2. Compute $\bar{\Phi}(\mathbf{c}_A)$ and $\bar{\Phi}(\mathbf{c}_G)$ by Equation (3.2) 3. As a representation, choose \mathbf{c}_G if $\bar{\Phi}(\mathbf{c}_A) > \bar{\Phi}(\mathbf{c}_G)$, \mathbf{c}_A otherwise. |
|--|

Figure 3.2. Geometric selection algorithm for representing multiple documents in the Riemannian manifold based on the Fisher information metric

Table 3.1. Test collections.

	AP	WSJ	GOV2
TREC topics	51-200	51-200	701-800
#docs	242,918	173,252	25,205,179

For the experiments, we use 3 standard collections from TREC³. Table 3.1 shows the statistics of the collections. To estimate a language model from each document, we use the Dirichlet smoothing. For each task, the initial results are obtained by query-likelihood scores which are computed under an independence assumption as follows:

$$Pr(Q|D) = \prod_{q \in Q} Pr(q|D)$$

where $Pr(q|D)$ is estimated by Equation (3.3).

For index building, we used the Indri system [126]. Each document was stemmed by the Krovetz stemmer and stopped by a standard stopword set [41]. To test the significance of results, we performed a randomization test [122].

Note that further discussions about the geometry of multiple documents can be found in Appendix A and A.

3.4.1 Cluster Retrieval

Cluster retrieval involves finding the best document cluster [74, 81]. We first retrieve the top 100 documents for each query according to query-likelihood scores.

³<http://trec.nist.gov/>

Next, we perform k NN clustering [67]. That is, assuming that each returned document is a cluster centroid, a cluster is formed by its $k-1$ nearest neighbors (k is set to 5). We use cosine similarity as a similarity measure. In fact, since cosine similarity assumes the Euclidean metric space, other similarity measures may perform better for our representation technique which assumes a different metric. However, since arbitrary clusters are assumed in cluster retrieval, we use the same similarity measure as used in previous work [81].

Once we have clusters, we represent each cluster by the arithmetic mean of language models of documents in a cluster assuming the Euclidean metric. On the other hand, assuming the Fisher information metric, we can determine a representation via geometric selection between the arithmetic mean and the normalized geometric mean of the documents.

Note that computing the normalization factor for the normalized geometric mean may look tricky because we have to sum the geometric means for all unique terms in a corpus. However, we can easily compute it as follows. V and V^* denote the vocabulary of a corpus and a set of multiple documents to be considered, respectively. Then, the normalization factor can be written as follows:

$$\sum_{w \in V^*} \prod_{i=1}^k \text{Pr}(w|D_i) + \sum_{w \in (V-V^*)} \prod_{i=1}^k \text{Pr}(w|D_i)$$

The first term does not matter because usually there are not so many terms in a set of multiple documents used many IR applications. In the second term, assuming the Dirichlet smoothing, $tf_{w,D_i} = 0$ because w is a term which does not appear in the set of documents. Hence, the second term is rewritten in a computable form as follows:

$$\begin{aligned} & \sum_{w \in (V - V^*)} \left(\prod_{i=1}^k \frac{\mu \cdot cf_w / |C|}{|D_i| + \mu} \right)^{1/k} \\ &= \mu \frac{|C| - \sum_{w \in V^*} cf_w}{|C|} \left(\prod_{i=1}^k \frac{1}{|D_i| + \mu} \right)^{1/k} \end{aligned}$$

Evaluation of various representation techniques such as concatenation or Comb-Max [42] for cluster retrieval has been already done by Liu and Croft [81]. They concluded that the geometric mean representation outperforms other techniques. Therefore, we do not intend to repeat the same work. Instead, we focus on geometric interpretations for experimental results.

For a fair comparison, the same clusters are given to each representation technique. The only parameter to be tuned is the smoothing parameter for the initial results. We set the parameter so that Mean Average Precision (MAP) for the initial results by the query-likelihood $Pr(Q|D)$ is maximized. Evaluation is performed using all topics. Since our goal is to find the best cluster, we use Precision at 5 (P@5) in order to evaluate the cluster first ranked by each representation technique, i.e. how many relevant documents the cluster has. Table 3.2 shows the results. In addition to the arithmetic mean and geometric selection, we present results using the geometric mean as well.

For all collections, representations by the geometric mean and geometric selection show better performance than representations by the arithmetic mean. Except for GOV2, The improvements are statistically significant. These experiments indicate some interesting points. First, in geometric selection, the normalized geometric means were selected as representations which minimize the Fréchet sample function for all queries across all collections. In other words, the normalized geometric means are better approximations to the Fréchet sample mean. Second, since the normalized geometric means selected by geometric selection lead to consistently better retrieval results, we may say that the goodness of a representation for this task is related

Table 3.2. Results for cluster retrieval. A-MEAN, G-MEAN and SELECT mean representations by the arithmetic mean, by the geometric mean, and by geometric selection, respectively. The numbers are P@5 scores. A * indicates a statistically significant improvement over A-MEAN ($p < 0.05$).

	AP	WSJ	GOV2
A-MEAN	0.3053	0.4747	0.5374
G-MEAN	0.3347*	0.5040*	0.5576
SELECT	0.3347*	0.5027*	0.5556

to how close the representation is to the center of mass, i.e. the Fréchet sample mean. Moreover, this justifies the assumption of the geometry defined by the Fisher information metric. Lastly, since geometric selection does not consider the geometric mean but the normalized geometric mean, the results in the ‘SELECT’ row are exactly the same as those by the normalized geometric means. Therefore, the differences between the ‘G-MEAN’ row and the ‘SELECT’ row are caused by the normalization. As you see, since the differences are small, we suggest that the geometric mean without normalization can be a better choice in practice.

3.4.2 Pseudo-Relevance Feedback

Lavrenko and Croft’s relevance model [71] is one of the standard language modeling approaches for pseudo-relevance feedback. The model assumes that the top k retrieved documents for query q are sampled from an underlying relevance model for q . That is, a hidden multinomial model relevant to a user information need exists, and we estimate the model from the top k documents. Then, we sample terms which describe the information need better than the original query and use the terms for query expansion.

Estimation of the relevance model is done by the following formula:

$$Pr(w|q) = \frac{\sum_{i=1}^k p(w|D_i)Pr(q|D_i)Pr(D_i)}{p(q)} \quad (3.12)$$

where q is a user query, w is a candidate for expansion terms, and D_i is a document in the top k initial results, respectively.

Although this is derived from a Bayesian model, we can see this as a representation for the top k documents by the arithmetic mean rewriting Equation (3.12) as follows:

$$\sum_{i=1}^k p(w|D_i) \frac{Pr(q|D_i)Pr(D_i)}{p(q)} = \sum_{i=1}^k p(w|D_i)Pr(D_i|q)$$

This has the same form as the weighted arithmetic mean of Equation (3.7). In other words, $Pr(w|D_i)$ is a multinomial parameter and $Pr(D_i|q)$ represents a distribution over a sample space limited by q , i.e., \hat{Q} . In the standard implementation of the relevance model by the Indri system [126], $Pr(D)$ is assumed to be uniform. Hence,

$$Pr(D_i|q) = \frac{Pr(q|D_i)Pr(D)}{\sum_{i=1}^k Pr(q|D_i)Pr(D)} = \frac{Pr(q|D_i)}{\sum_{i=1}^k Pr(q|D_i)}$$

That is, the weight $\hat{Q} = Pr(D_i|q)$ is the normalized query-likelihood scores obtained in the initial retrieval phase. Therefore, we can say that the relevance model represents a group of the top k documents combining the language models by the arithmetic mean weighted by the initial search results. In this sense, we can say that the relevance model implicitly assumes the Euclidean metric space.

We can replace the arithmetic mean by the normalized geometric mean to develop a new representation as follows:

$$Pr(w|q) = \frac{\prod_{i=1}^k p(w|D_i)^{Pr(D_i|q)}}{\sum_{w \in V} \prod_{i=1}^k p(w|D_i)^{Pr(D_i|q)}} \quad (3.13)$$

We can consider the original relevance model and this model as two approximated representations in the Riemannian manifold defined by the Fisher information metric. To determine a representation, we use geometric selection and call the selected model the “geometric relevance model”.

Table 3.3. Results for pseudo-relevance feedback. RM and GRM mean the relevance model and the geometric relevance model, respectively. The numbers are MAP scores. A * indicates a statistically significant improvement over RM ($p < 0.01$).

	AP	WSJ	GOV2
RM	0.2541	0.3531	0.3204
GRM	0.2769*	0.3851*	0.3300*

We compare the geometric relevance model with the relevance model. For each query, we first retrieve the top k documents by query-likelihood scores and build a relevance model or geometric relevance model for the documents. Then, we choose the top M terms according to probabilities of the terms in the models. Finally, we expand the original query combining the expansion terms using an interpolation weight λ in the Indri query language. The parameters k , M and λ are tuned so that MAP scores by the relevance model are maximized. The same parameters are used for the geometric relevance model. Topic 51-150 for AP and WSJ and topic 701-750 for GOV2 are used as training topics to learn the parameters. Topic 151-200 for AP and WSJ and topic 751-800 for GOV2 are used as test topics. We retrieve up to 1000 results for each expanded query and use MAP as the evaluation metric.

Table 3.3 shows the results. The geometric relevance model significantly outperforms the relevance model for all three collections. Similar to cluster retrieval, geometric selection selected models by Equation (3.13) rather than the original relevance model as representations for all queries except for three queries of GOV2. That is, the geometric mean is a better approximation to the center of mass for this task. This provides more empirical evidence that the geometric mean can be an appropriate choice for representation.

3.5 Conclusions

In this chapter, we showed that using Information Geometry, the arithmetic mean and the normalized geometric mean are approximation points to the center of mass

in the Euclidean space or in a statistical manifold. In particular, through empirical evidence from experiments for various generic IR tasks, we demonstrated that the normalized geometric mean is closer to the center in the statistical manifold, which often leads to better retrieval performance. Based on these results, we will use geometric-mean based representations as a primary technique for combining multiple contexts derived from various social media structures.

CHAPTER 4

HIERARCHICAL STRUCTURES AND BLOG SITE SEARCH

Hierarchical structures are explicit in most social applications. For example, a blog site has many postings. Also, a forum has many threads. In turn, each thread contains many postings. These relations by ownership or containments make hierarchical structures. Of course, we may consider more implicit hierarchical structures in social applications, e.g., hierarchical structures by clustering, hierarchical structures by concept-term relations, etc. However, we here consider only explicit hierarchical structures defined by ownership. Based on this definition, we introduce techniques to leverage hierarchical structures in social applications. In particular, to demonstrate how hierarchical structures can be exploited for retrieval tasks, we present a blog site search task because this is one of the most relevant tasks that can benefit from the exploitation of hierarchical structures. Via blog site search, we introduce how to extract relevant contexts considering hierarchical structures and how to make representations for the contexts.

4.1 Blog Site Search

Blog site search is to identify relevant blog sites. For example, when selecting blogs to subscribe through RSS or ATOM, it would be more effective to find blogs which cover mostly the topic of interest than to find blogs which contain a few relevant postings. Further, many blogs address a small number of specific topics rather than being completely general. If there is a relevant blog related to a specific topic, then

that blog can be expected to consistently generate good quality postings about the topic. The creation of Blog Distillation Task of the TREC 2007 Blog track [84] whose goal is finding a feed with a “principle, recurring interest in a topic”, reflects the interest in this type of search.

In this chapter, we focus on search techniques for complete blogs rather than postings. Since the term “blog search” often means “posting search” we instead use the term “blog site search”, where a blog site refers to the collection of postings in the blog.

As an example of the difference between blog site and blog posting searches, consider the following two queries:

Q1: “iPad review”

Q2: “mobile gadget review”

In the case of Q1, the user has specified a product name and probably expects to retrieve postings reviewing that product. Generally, blog sites containing reviews about only one product are rare and such reviews are scattered over many review sites. Therefore, Q1 would be better handled using posting search. On the other hand, Q2 is more general. Although it would be difficult for a single posting to include all the content relevant to Q2, a set of postings, i.e., a blog site, can address a general topic. Q2 is appropriate for blog site search, and is more likely to lead to a subscription to a feed.

4.2 Blog Site Representations Using Hierarchical Contexts

A blog site consists of its postings. The relation between a posting and a blog site, e.g., an ownership or authorship, establishes a hierarchical structure. Based on this hierarchical structure, we can consider two ways of extracting relevant contexts.

The first method is to consider all the postings in a blog site and make a context called “global context”. That is, this context is independent of user queries. Accord-

ingly, this context can usually produce a smooth topic distribution and reflect overall topics addressed in the blog site. On the other hand, there is a possibility that locally distributed information or topics represented by individual postings is lost.

The second method is to consider several relevant postings in a blog site and make a context. This is called “local context” and dependent on user queries because postings are selected according to the relevance to a query. This context preserves local information addressed in even a small subset of a blog site. However, there is a risk that representations based on these local contexts can be biased toward to the selected postings.

Considering these two contexts, we introduce three representation techniques. One technique is based on a global context whereas other two techniques are based on local contexts. Specifically, we consider resource selection techniques in distributed information retrieval [19], which are used to select the most relevant collections from a large number of possible collections. That is, resource selection is a representative technique using hierarchical structures. Since a blog site is a collection of postings and our target is finding relevant blog sites, our task is similar to resource selection. Therefore, the three representation techniques in this section are inspired by existing resource selection techniques. Among them, the first two techniques, i.e., global representation and query generation maximization are considered as baselines because they are blog site search adaptations of existing resource selection techniques. On the other hand, although the last technique, i.e., pseudo-cluster selection is also inspired by resource selection, we employ our geometric representation technique for representing multiple documents introduced in Chapter 3 because we need to make a blog site representation with multiple postings in a local context.

4.2.1 Global Representation

One of the simplest approaches for hierarchical structures treats a collection as a single, large document. That is, this approach is based on a global context. This approach has been widely used for resource selection in distributed information retrieval [19, 135]. For a blog site search, we can generate a virtual document for a blog site by concatenating all postings in a blog. This virtual document D_i for a blog site c_i can then be represented using a language model (probability distribution of words) and the query likelihood of the document for a query Q is used as a ranking function.

$$\begin{aligned}\Gamma_{GR}(Q, c_i) &= Pr(Q|D_i) \\ &= \prod_{q \in Q} Pr(q|D_i) \\ &= \prod_{q \in Q} \frac{tf_{q,D_i} + \mu \cdot cf_q/|C|}{|D_i| + \mu}\end{aligned}$$

where q is a query term of query Q , tf_{q,D_i} is the number of times term q occurs in virtual document D_i , $|D_i|$ is the length of virtual document D_i , cf_q is the number of times term q occurs in the entire collection, $|C|$ is the length of the collection, and μ is a Dirichlet smoothing parameter [139].

This simple, intuitive method was effective in TREC 2007 blog distillation task without any help from advanced techniques [32, 112]. Since the blog distillation task is very similar to blog site search, this method can be considered as a strong baseline. However, this technique has some problems. One of the problems is that the virtual document might be a mixture of various topics. In this case, it is hard for a single language model to accurately reflect the content of the blog site. Further, the content of the virtual document can be skewed by a few large postings.

Since this technique can capture a global context through a coarse-grained representation, we call this technique “global representation” and use it as a baseline for our experiments.

4.2.2 Query Generation Maximization

Si and Callan introduced a state-of-the-art technique for resource selection based on estimating the probabilities of relevance of documents in the distributed environment [120]. This method, which is referred to as “unified utility maximization”, does resource selection to maximize a utility function.

The utility function can be defined as a solution of two types of maximization problems. One is for high-recall and the other is for high-precision. Since our goal is finding relevant collections rather than relevant postings, we consider the high-recall case. The utility function for the high-recall problem is defined as follows:

$$U(\vec{\sigma}) = \sum_{i=1}^{N_C} I(c_i) \sum_{j=1}^{\tilde{n}_i} \tilde{R}(d_{ij})$$

where c_i is a collection, i.e., $\{d_{i1}, d_{i2}, \dots\}$, N_C is the number of total collections, \tilde{n}_i is the number of the returned documents from the collection c_i and $I(c_i)$ is an indicator function which is 1 if c_i is selected and 0 otherwise. $\vec{\sigma}$ is a selection vector, i.e., $[I(c_1), I(c_2), \dots, I(c_{N_C})]$ and $\tilde{R}(d_{ij})$ is an estimated probability of relevance of the returned document d_{ij} . As mentioned above, our goal is finding a selection vector to maximize the utility function with the limited number of selection; thus, the problem is described as follows:

$$\begin{aligned} \vec{\sigma}^* &= \arg \max_{\vec{\sigma}} \sum_{i=1}^{N_C} I(c_i) \sum_{j=1}^{\tilde{n}_i} \tilde{R}(d_{ij}) \\ &\text{subject to: } \sum_{i=1}^{N_C} I(c_i) = N_{\vec{\sigma}} \end{aligned} \quad (4.1)$$

where $N_{\vec{\sigma}}$ is the predetermined number for selection. The optimized solution of this problem is selecting $N_{\vec{\sigma}}$ collections with the largest expected number of the relevant documents, i.e., $\sum_{j=1}^{\tilde{n}_i} \tilde{R}(d_{ij})$.

In order to apply this method to blog site search, we simplify the process as follows. We build an index of postings ignoring which blog site the postings are from. Since we already know statistics of each collection, we can directly translate the query likelihood score to the probability of relevance of the document for a given query without any estimation process. Therefore, by substituting a query likelihood score for the probability of relevance, $R(d_{ij})$, we can rewrite Equation (4.1) as follows:

$$\vec{\sigma}^* = \arg \max_{\vec{\sigma}} \sum_{i=1}^{N_C} I(c_i) \sum_{j=1}^{\tilde{n}_i} P(Q|d_{ij})$$

where $P(Q|d_{ij})$ is the query likelihood of the document d_{ij} for the query Q as follows.

$$\begin{aligned} Pr(Q|d_{ij}) &= \prod_{q \in Q} Pr(q|d_{ij}) \\ &= \prod_{q \in Q} \frac{tf_{q,d_{ij}} + \mu \cdot cf_q / |C|}{|d_{ij}| + \mu} \end{aligned}$$

In this case, the optimized solution is selecting $N_{\vec{\sigma}}$ collections with the highest expected generation of the query, i.e., $\sum_{j=1}^{\tilde{n}_i} Pr(Q|d_{ij})$.

We induce a ranking function based on the maximization.

$$\Gamma_{QGM}(Q, c_i) = \sum_{j=1}^{\tilde{n}_i} Pr(Q|d_{ij})$$

Therefore, what we need to do is simply sum the query likelihood scores of postings from the same blog site in the ranked list which is returned from the index. Next, we can obtain a final ranked list in decreasing order of the sum value. It means that this method can be easily implemented by a simple post-processing after posting search. Since this representation is based on representations of individual postings and uses only postings in a ranked list, we can say that this is a local context-based method.

We call this modified method “query generation maximization” and use it as the second baseline for our experiments.

4.2.3 Pseudo-Cluster based Selection

Xu and Croft [136] showed that distributed information retrieval using clustering is very effective because clustering redistributes documents in collections and makes topic-based sub-collections. There are two methods to use clustering for distributed information retrieval. One is the global clustering method. It makes clusters using all documents regardless of the collection. The other is the local clustering method. It makes clusters using documents within a collection. After clustering, both of the methods build an index for each cluster and retrieve documents from relevant clusters.

However, since our goal is not to find relevant documents using resource selection but to find resources themselves, redistribution of documents of each collection using clustering is not likely to be effective. Instead, we create “pseudo-clusters” by ranking blog postings and then grouping highly-ranked postings from the same blog. To represent the pseudo-clusters, we employ the geometric-mean based representation technique discussed in Chapter 3 as follows:

$$Pr(w|g) = \left(\prod_{j=1}^{N_g} Pr(w|d_j) \right)^{\frac{1}{N_g}}$$

where w is a word, g is a cluster, d_j is a document in cluster g , and N_g is the number of documents in cluster g . The geometric mean is relatively robust against the situation where the influence of some documents overwhelms that of the others.

If we apply the representation method to our pseudo-cluster, then we can easily compute a query likelihood of blog site c_i by a geometric mean of query likelihoods of postings of blog site c_i in the ranked list (under a unigram assumption) as follows.

$$\begin{aligned}
Pr(Q|c_i) &= \prod_{q \in Q} Pr(q|c_i) \\
&= \prod_{q \in Q} \left(\prod_{j=1}^{\tilde{n}_i} Pr(q|d_{ij}) \right)^{\frac{1}{\tilde{n}_i}} \\
&= \left(\prod_{j=1}^{\tilde{n}_i} \left(\prod_{q \in Q} Pr(q|d_{ij}) \right) \right)^{\frac{1}{\tilde{n}_i}} \\
&= \left(\prod_{j=1}^{\tilde{n}_i} Pr(Q|d_{ij}) \right)^{\frac{1}{\tilde{n}_i}}
\end{aligned} \tag{4.2}$$

Note that the number of documents from each blog site in the ranked list is different in contrast to Liu and Croft's original method using actual clustering. Although query generation maximization also assumes different numbers of documents for blog sites, it looks reasonable that blog sites having more relevant postings, i.e., more documents in the ranked list get good scores. On the other hand, in case of representation by a geometric mean, this causes a problem. For example, a blog site p has a single document in a ranked list and the document is ranked at the second place, whereas a blog site q has three documents in the ranked list, which are ranked at the first, the third and the fourth places. In this case, blog site p might have a higher geometric mean than q . This seems unfair. To resolve this, we use K documents with high ranks in the ranked list regardless of the number of documents of each blog site, where K is a parameter independent of clusters. Then, our ranking function is defined as follows.

$$\Gamma_{PCS}(Q, c_i) = \left(\prod_{j=1}^K Pr(Q|d_{ij}) \right)^{\frac{1}{K}}$$

If a blog site has less than K documents in the ranked list, then we can estimate the upper bound of the geometric mean of the blog site using the minimum query likelihood score in the list as follows.

$$d_{\min} = \arg \min_{d_{ij}} Pr(Q|d_{ij})$$

$$\Gamma_{PCS}(Q, c_i) = \left(Pr(Q|d_{\min})^{K-\tilde{n}_i} \prod_{j=1}^{\tilde{n}_i} Pr(Q|d_{ij}) \right)^{\frac{1}{K}} \quad (4.3)$$

This can be also simply computed from the ranked list of postings. We refer to this method as “pseudo-cluster selection”. This technique is also classified as a local context-based representation.

4.3 Experiments

4.3.1 Data

We used the TREC Blogs06 Collection [83] for experiments. The collection was crawled by the University of Glasgow from December 6, 2005 to February 21, 2006 and contains 3,215,171 postings and 100,641 unique blog sites. Since our approaches are based on the postings, we used only posting components in the collection. The postings were stemmed by the Porter stemmer after HTML tags were removed.

We made new relevance judgments for blog site search for ourselves. We selected 50 queries from queries of the topic distillation task of the TREC 2002 Web Track and the TREC 2003 Web Track. The queries of the topic distillation task are a mixture of abstract queries and explicit queries, and we felt that they fit well with the experiments.

To make the relevance judgments for each query, we used a pooling method [125]. Three techniques introduced in Chapter 4.2, relevance feedback [5] and dependence models [89] contributed to the pools. As a result, we made judgments for about 2,500 blog sites. The criteria used for relevance is as shown in Table 4.1.

In the second set of experiments, we used the data for the TREC 2007 blog distillation task.

Table 4.1. The criteria for the relevance judgments.

Grade	Criterion
0	The blog site does not consistently create postings relevant to the topic.
1	More than 25% of the postings in the blog deal with the topic.
2	More than 50% of the postings in the blog deal with the topic.
3	More than 75% of the postings in the blog deal with the topic.

4.3.2 Experimental Design

We do experiments for three blog site representation techniques, i.e., global representation, query generation maximization and pseudo-cluster selection.

For global representation, we built an index of each blog site after concatenating each posting from the same blog site. We used the query likelihood retrieval model as the ranking method for the global representation. Query generation maximization and pseudo-cluster selection require an initial retrieval. We built an index from all postings and used the query likelihood retrieval model for the initial run. To get the result, we post-processed the results of the initial run by using the respective technique.

For our experiments, we used Indri [126] as the retrieval system. The randomization test was performed to test statistical significance of improvements of retrieval results.

We performed exhaustive grid search to find optimal parameters for each technique. In case of the global representation, we have one parameter to be trained, i.e., the μ parameter for Dirichlet smoothing. The query generation maximization requires training for two parameters, i.e., the smoothing parameter and the number of the documents to be used for the post-process of the results of the initial retrieval, N_R . For the pseudo-cluster selection, the parameter for the cluster size restriction, K , is additionally required. We used the normalized discounted cumulative gain (NDCG) [55], the mean average precision (MAP) and the precision at the rank 10 (P@10) as the evaluation measures. For binary relevance judgment-based metrics such as MAP

and P@10, we regarded a blog site having a grade of Table 4.1 equal to or greater than 1 as a relevant blog site. The parameter trainings were also done for each measure.

We performed 10-fold cross validation in order to evaluate performance. 50 queries were randomly partitioned. For one partition, the parameters were trained with all the other partitions and performance for the partition is evaluated with the trained parameters. We concatenated the ranked lists from each partition and evaluated them.

4.3.3 Results

Table 4.2 presents the performance of each representation method. Two baselines, global representation and query generation maximization showed similar performance. Although pseudo-cluster selection is proposed to use the geometric mean for representing pseudo-clusters according to the conclusion of Chapter 3, we employed the arithmetic mean method as well as the geometric mean method to make sure if the geometric representations actually work.

As you see, when using the geometric mean, pseudo-cluster selection significantly outperformed the other techniques. On the other hand, the arithmetic mean-based pseudo-cluster selection does not work well. Also, we conducted the geometric selection algorithm of Chapter 3 as well. The algorithm selected the geometric means as an approximated Fréchet mean for all queries. These results support our geometric representation. Note that from now on in this thesis, pseudo-cluster selection refers to the technique using the geometric mean if any specific method is not mentioned to avoid confusion.

An interesting observation is that local context-based methods show relatively better performance than a global context-based method, i.e., global representation. This may be because global representation collapses the hierarchical structures. However, as you see the performance differences among these local context-based methods,

Table 4.2. Retrieval performance by blog site representation techniques. α and β in a cell indicate statistically significant improvement over the baselines, global representation and query generation maximization, respectively. ($p < 0.1$)

	NDCG@100	MAP	P@10
Global Representation	0.5448	0.3708	0.2780
Query Generation Maximization	0.5422	0.3785	0.2920
Pseudo-cluster Selection (geometric mean)	0.5632	0.4091 $^{\alpha\beta}$	0.3300 $^{\alpha\beta}$
Pseudo-cluster Selection (arithmetic mean)	0.5553	0.3961 $^{\alpha}$	0.3180

we should be careful when combining this hierarchical structural evidence. That is, proper methods such as pseudo-cluster selection should be employed.

In a practical sense, query generation maximization and pseudo-cluster selection have an advantage over global representation. Nowadays, most of the blog publishing or blog search service providers have already provided posting search services. Since the two techniques use the results of posting search, they can be easily implemented by reusing the index for posting search.

4.4 Incorporating Global Contexts

We showed that the performance gain is larger when using local contexts, i.e., a few relevant postings than when using a global context, i.e., the whole blog site. However, we cannot ignore such a global context because it may have a potential to lead to better blog site search. In this chapter, we discuss which kinds of blog sites can be recognized as relevant, and suggest a way of incorporating global contexts into our local context-based techniques.

4.4.1 Types of Blog Sites

In order to better understand the problem of blog site retrieval, we classified blog sites into three types based on how they are managed and the degree of diversity of the topics covered.

Type I is the diary type of blog. In this type, a blogger usually posts descriptions of their daily life. In many cases, the postings are related to personal issues such as relationships, appointments or concerns. Some postings can be about a person's interests or opinions about a specific issue or object. However, it is rare that other postings about similar topics are regularly updated in the blog site.

Type II is the news blog. Documents covering a large number of topics are posted, and many of these blogs are managed by an organization or a company. Another common situation is when most of the postings are not composed by the blogger but are collected by the blogger. For example, if a blogger finds some good articles while surfing news sites, they may copy and paste them into their own blog. In this case, the blog functions like a scrapbook, which causes many duplicate documents over the whole web collection. In sum, even though this type contains relatively good quality documents, it often lacks originality and is not topic-centric.

Furthermore, this second type is related to an important issue of blog search. Blogs are a subset of general Web pages. When blog search services crawl the Web to find blog postings, they typically identify them by checking whether the Web page contains a feed link for RSS or ATOM. Many general Web news sites also contain feed links for their subscribers, and this can cause these sites to be included in the blog collection. Since such sites have not only a large number of good quality documents but also relevant documents for all kinds of topics, they may often be retrieved. To prevent this requires some type of penalty factor.

Type III is the topic-focused type of blog. This is managed by one or a few individuals and concentrates on a small number of topics. The quality of postings varies on the blogger, but often is good. This type of blog site with a topic specialty exists for many topics. The typical examples that are frequently seen are product review blogs or political advocate blogs. It is probable that documents related to the specific topic are regularly posted in this type of blog site. The success of our

Table 4.3. Type classification of blog sites

Type	Topic-centric	Document Quality	Originality
Type I (Diary)	Low	Low	High
Type II (Newspaper)	Low	High	Mid
Type III (Topic-focused)	High	Mid	High

retrieval methods will depend on how well we are able to find this type of blog site for a given query. Table 4.3 summarizes the properties of each type.

To verify the validity of our categories, we manually classified 100 blog sites randomly selected from the pools for relevance judgments. Of course, it is not easy to simply classify a blog site into a single category because diary postings, news postings and topic-focused postings might coexist in a blog site. For this reason, we classified them by observing what type of postings mainly exists in the blog site. There were some cases that we could not decide which category a blog site is in because it did not match any category. Most of such blog sites were spam sites, e.g., sites which do not contain real contents but instead are mostly advertisement links. We tagged such sites as “Unclassifiable”.

Three annotators independently labeled the blog sites. By majority voting, we assigned the label which more than two annotators agreed with to each blog site. If all annotators had different labels for a blog site, then we tagged the site as “Unclassifiable”. Table 4.4 presents the result. Most blog sites were mapped onto our categories. As we expected, the majority of relevant blog sites were in the topic-focused category. To measure inter-annotator agreement, Fleiss’ κ [40] was computed. The coefficient was 0.76 and this indicates a substantial agreement.

Table 4.4. Manual classification result with 100 blog sites

Type	#Blog Sites	#Relevant Blog Sites
Unclassifiable	7	0
Type I (Diary)	26	2
Type II (News)	25	1
Type III (Topic-focused)	42	11

4.4.2 Diversity Penalty

Based on the previous subsection, we need to penalize Type I and Type II blog sites. To do this, we focus on the fact that they are not topic-centric. Accordingly, we considered a method for penalizing blog sites with diverse topics.

We have to decide whether or not the blog site is topic-centric at the global level, i.e., the blog site level. Therefore, the penalty should be able to be used at the global level. Further, it will be more helpful if the penalty can reflect the relevance for the topic.

4.4.2.1 Diversity Penalty by Global Representation

We already have seen a component that could be used as a diversity penalty. It is the query likelihood score from the global representation used as a baseline in Chapter 4.2. We compute the score at the global level. Further, if the blog site deals with the diverse topics, then the distribution of the words in the blog site are probably widely scattered. As a result, the occurrence of the words closely related to a specific topic might be relatively low compared to the topic-centric blog sites. It causes a low query likelihood score in the language modeling-based retrieval.

Figure 4.1 shows indirect evidence supporting this claim. We obtained the result ranked list for 50 queries by using each blog site representation technique. We analyzed the distribution of the number of postings in the returned blog sites according to the above mentioned techniques. Further, we provide the distribution of the number of postings of blog sites in the entire collection by randomly selecting the same number of blog sites as those in the ranked lists (“Random” in Figure 4.1). As we

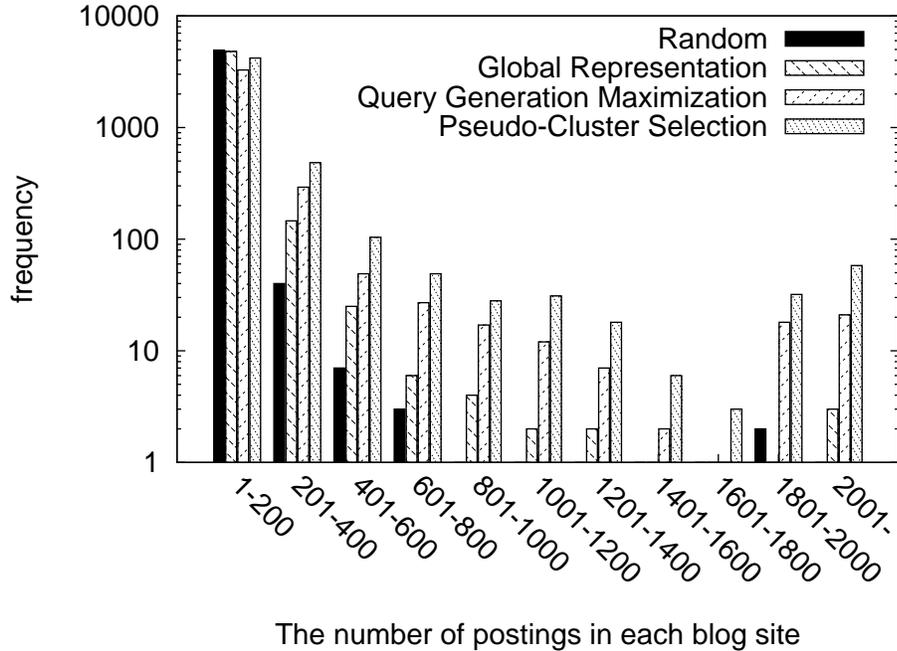


Figure 4.1. The distribution of the number of postings in the blog sites returned by each blog site representation technique

can see from the histogram in Figure 4.1, the global representation definitely returned much fewer blog sites which have a large number of postings. Although it is an over-generalization to assume that a blog site having many documents is diverse, there is such a tendency. For example, it is apparent that the news sites where thousands of articles are posted daily have much more documents than the topic-focused blogs where at most several postings a week are registered.

In summary, the query likelihood score can be useful as a measure of diversity of blog sites. Furthermore, this score reflects the relevance of the blog site for the given topic. Accordingly, to supplement the other two blog site representation techniques, we can use this score as a penalty factor for diversity by multiplying it by the previous ranking function as follows.

For query generation maximization,

$$\Gamma_{QGM-GR}(Q, c_i) = \Gamma_{QGM}(Q, c_i)^{1-\pi} \cdot \Gamma_{GR}(Q, c_i)^\pi$$

For pseudo-cluster selection,

$$\Gamma_{PCS-GR}(Q, c_i) = \Gamma_{PCS}(Q, c_i)^{1-\pi} \cdot \Gamma_{GR}(Q, c_i)^\pi$$

where π is a weight parameter. The multiplication is used to prevent from being biased as in Chapter 4.2.3. Further, it can be interpreted as a linear combination of the log probabilities as well as a type of smoothing.

4.4.2.2 Clarity Score as a Penalty Factor

Another candidate which we can consider as a penalty factor for diversity is a clarity score. Cronen-Townsend et al. [29] showed that query performance can be predicted using the relative entropy between a query language model and the corresponding collection language model as a clarity score. That is, since the query which has the similar language model to that of the collection seems somewhat ambiguous, we do not expect good retrieval performance with that query.

However, in our work, we want to know the difference between a blog site and the whole collection rather than between a query and a collection. We assume that if a blog site covers many general topics, then the language model of the blog site is similar to that of the whole collection. On the other hand, in a blog site which addresses a few specific topics, some terms related to the topics occur relatively frequently and accordingly, the language model is expected to be different from that of the whole collection. Thus, we compute the clarity score by using the relative entropy, or Kullback-Leibler divergence [27] between a blog site and the whole collection as follows.

$$\text{Clarity}(c_i) = \sum_w Pr(w|c_i) \log \frac{Pr(w|c_i)}{Pr(w|\text{Coll})}$$

We also use this score as a penalty factor for diversity by multiplying it by the previous ranking function as follows.

For query generation maximization,

$$\Gamma_{QGM-Clarity}(Q, c_i) = \Gamma_{QGM}(Q, c_i)^{1-\pi} \cdot \text{Clarity}(c_i)^\pi$$

For pseudo-cluster selection,

$$\Gamma_{PCS-Clarity}(Q, c_i) = \Gamma_{PCS}(Q, c_i)^{1-\pi} \cdot \text{Clarity}(c_i)^\pi$$

4.4.2.3 Diversity Penalty by Random Sampling

We need to keep additional information like the index for global representation in order to use two penalty factors introduced above because they depend on the statistics of a whole blog site. This requirement might be a considerable burden for most blog service providers. Further, the penalty factors ignore boundaries of postings, and accordingly, there can be bias problems. As seen in Figure 4.1, the global representation is biased toward small size blog sites. Both penalty factors favor collections which have long postings because such long postings dominate the whole blog site, regardless of the number of them, and the blog sites are considered topic-centric.

To address these problems, we suggest a randomized approach. In pseudo-cluster selection, we use postings in the ranked list to get postings relevant to a given topic. On the other hand, we randomly sample M postings from a blog site to obtain postings independent of any topic. Note that the randomly sampled postings might or might not be in the ranked list. We compute the query likelihoods for the sampled postings with the given query. If the blog site is topic-centric and relevant to the topic, then the postings are likely to be relevant to the topic and the query likelihoods have high values. Otherwise, postings about various topics are picked and the query likelihoods have small values. Therefore, the query likelihoods can be used for estimating diversity

of a blog site. Further, this approach is free from bias problems in that postings are directly used, and additional information is not required.

We make a diversity penalty factor with the query likelihoods of the randomly sampled postings in the same way as used in pseudo-cluster selection. In other words, we compute a geometric mean of the query likelihoods. This diversity penalty factor can be used by multiplying it by the previous ranking function as follows.

For query generation maximization,

$$\Gamma_{QGM-Random}(Q, c_i) = \Gamma_{QGM}(Q, c_i)^{1-\pi} \cdot \left(\prod_{j=1}^M Pr(Q|r_{ij}) \right)^{\frac{\pi}{M}}$$

For pseudo-cluster selection,

$$\Gamma_{PCS-Random}(Q, c_i) = \Gamma_{PCS}(Q, c_i)^{1-\pi} \cdot \left(\prod_{j=1}^M Pr(Q|r_{ij}) \right)^{\frac{\pi}{M}}$$

where r_{ij} is the j^{th} randomly selected posting of blog site c_i .

Note that this diversity penalty factor may look to be derived from local contexts in that this uses individual postings. However, while pseudo-cluster selection uses postings selected in a space limited by relevance scores, this penalty uses postings selected in an entire space of postings corresponding to a blog site. From this point of view, we can call this penalty a global context.

A problem of this random sample-based approach is that the retrieval result is changed every time even when there is not any change in the target collection. Such unstable search results might frustrate users. Therefore, a specific (pseudo-random) sampling may be more desirable than purely random sampling. The choice of a sampling method depends on the goals of blog site search services. If a blog site search service favors blog sites that have a more recent focus on a specific topic, then using M recent postings in each blog site instead of randomly sampled postings can

Table 4.5. Retrieval performance for blog site representation techniques combined with each penalty factor. GR, QGM and PCS stand for global representation, query generation maximization and pseudo-cluster selection, respectively. α and β in a cell indicate statistically significant improvement ($p < 0.1$) over the baselines, global representation and query generation maximization, respectively.

	NDCG	MAP	P@10
QGM with Penalty by GR	0.5344	0.3957	0.3040 $\alpha\beta$
PCS with Penalty by GR	0.5631 α	0.4217 $\alpha\beta$	0.3240 $\alpha\beta$
QGM with Penalty by Clarity	0.5286	0.3610	0.2760
PCS with Penalty by Clarity	0.5207	0.3444	0.2720
QGM with Penalty by Random Postings	0.5579 β	0.4011 $\alpha\beta$	0.3012
PCS with Penalty by Random Postings	0.5782 $\alpha\beta$	0.4213 $\alpha\beta$	0.3252 $\alpha\beta$
QGM with Penalty by Recent Postings	0.5705 β	0.4134 $\alpha\beta$	0.3080 $\alpha\beta$
PCS with Penalty by Recent Postings	0.5841 $\alpha\beta$	0.4323 $\alpha\beta$	0.3280 $\alpha\beta$

be a good choice. We provide experimental results in cases of using recent postings as well as randomly sampled postings in the next section.

4.4.3 Experimental Results

We did experiments to study the effectiveness of each suggested penalty factor using the same partitions for 10-fold crossvalidation. Table 4.5 shows the experimental results after applying the penalty factors.

The results show that there is the improvement in performance for both of the methods in case of using the global representation score as the penalty factor. In the experiment for the query generation maximization, the effectiveness according to MAP and P@10 became better, but the improvement was not still statistically significant except for P@10. In case of pseudo-cluster selection, the performance only for MAP was improved, whereas the performance for P@10 and NDCG was similar or lower compared to the original method. Nevertheless, the performance with respect to the baselines for both of the measures was statistically significantly improved.

In contrast, using the clarity score as a penalty factor hurt the overall performance. Although the degradation of the performance with respect to the baselines for all the measures was not statistically significant, the performance became consistently worse.

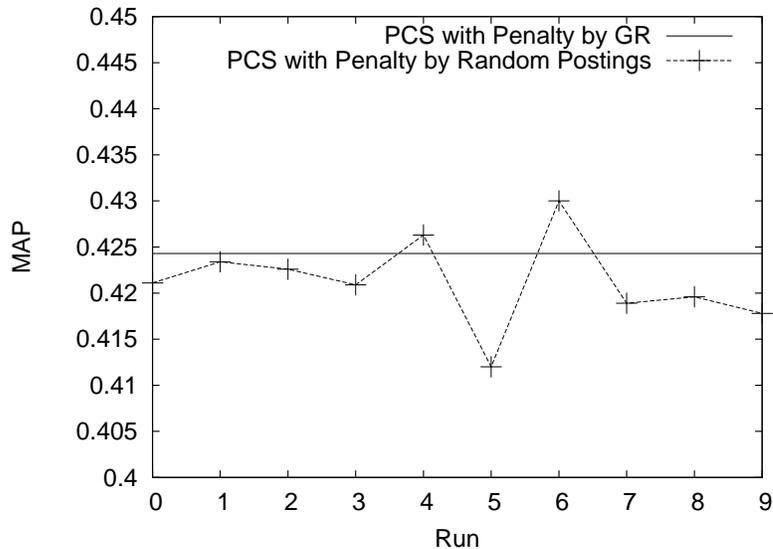


Figure 4.2. MAP scores for each run of pseudo-cluster selection with a penalty factor by random postings. GR and PCS stand for global representation and pseudo-cluster selection, respectively.

The reason is that the clarity score is independent of the queries and does not reflect the relevance for the topics at all. That is, even when we want a factor measuring diversity rather than relevance, the factor may need to reflect relevance to some extent.

Since the results by a penalty factor by randomly sampled postings are different every time, we did the same run 10 times and used the average of evaluation values for each query. Figure 4.2 shows the change of the MAP score for each run of pseudo-cluster selection with a penalty factor by random postings. Note that the scores have similar values to an MAP score of pseudo-cluster selection with a penalty factor by global representation. There was no statistically significant difference ($p < 0.1$) between the performance of pseudo-cluster selection with a penalty factor by global representation and the performance of each run of pseudo-cluster selection with a penalty factor by random postings.

Penalty factors by random sampling were very effective for both query generation maximization and pseudo-cluster selection. The methods consistently showed sub-

stantial performance gain for NDCG and MAP (and a small loss for P@10) compared to the original method. The improvement is statistically significant over baselines. In particular, a penalty factor by recent postings showed the best performance in our experiments. However, we cannot rule out the possibility that our relevance judgments are unconsciously biased toward recent postings of each blog site.

In summary, we can conclude that combining global contexts and local contexts presents more consistent improvements. That is, the more hierarchical structures considered, the better results we get. Thus, this result demonstrate evidence that hierarchical structures in blog sites can be helpful for the blog site search task.

4.5 Blog Distillation Task

The blog distillation task which was defined in TREC 2007 [84], identifies feeds relevant to a specific topic. The task is almost the same as blog site search in that a blog site generally has a feed and the feed is a summary of the blog site. Therefore, we can apply our blog site search techniques to the task.

The judgment set of TREC 2007 blog distillation contains 17,411 judged feeds for 45 topics. Although the distillation task is finding relevant feeds in the feed components in the TREC Blogs06 collection, we use only the posting collections as done before. Thus, we have to convert result blog site IDs to the feed IDs.

We applied the same baselines and the techniques that showed good performance in the previous experiments, i.e., global representation, pseudo-cluster selection and pseudo-cluster selection with a penalty factor by global representation, random postings and recent postings for penalizing diversity. We used parameters learned by our relevance judgments. Table 4.6 shows the results of experiments.

Surprisingly, global representation performed better than pseudo-cluster selection. We suspected that the reason is that pseudo-cluster selection is sensitive to query lengths. To confirm our assumption, we computed the correlation between the query

Table 4.6. Retrieval performance for the blog distillation task. GR, QGM and PCS stand for global representation, query generation maximization and pseudo-cluster selection, respectively. α and β in a cell indicate statistically significant improvement ($p < 0.1$) over the baselines, global representation and query generation maximization, respectively.

	MAP	P@10
GR	0.3454	0.4889
QGM	0.2709	0.4311
PCS	0.3171	0.4622
PCS with Penalty by GR	0.3725 $^{\alpha\beta}$	0.5356 $^{\alpha\beta}$
PCS with Penalty by Random Postings	0.3542 $^{\beta}$	0.5289 $^{\alpha\beta}$
PCS with Penalty by Recent Postings	0.3480 $^{\beta}$	0.5356 $^{\alpha\beta}$

length and the following performance differences of global representation and pseudo-cluster selection.

$$MD = \frac{MAP_{GR} - MAP_{PCS}}{MAP_{PCS}} \quad (4.4)$$

where MAP_{GR} and MAP_{PCS} are the Mean Average Precision (MAP) of the global representation and the pseudo-cluster selection, respectively.

Kendall’s τ was computed with MD and the number of terms in each query where p -value < 0.1 . The correlation coefficient value was about 0.2 and the result was statistically significant. Since the value is somewhat small, we cannot say that they are tightly correlated. Nevertheless, there is some relationship between them. That is, for the longer queries, pseudo-cluster selection can be better than global representation. This is not unreasonable. Since global representation uses a greater amount of text, other terms closely related to the topic but not in the query as well as the query terms can be often used in the relevant blog. That is, the effect of the terms in the query is diluted by the large amount of text. Consequently, if the query is long or it contains terms which are not generally used, then even a relevant blog might be determined to be irrelevant to the query. On the other hand, pseudo-cluster selection is a technique that represents a cluster with a relatively small number of

documents (In our experiments, $K = 5$). Here, the documents are directly selected by the initial search using the given query. Therefore, when the query is clear, pseudo-cluster selection works well. But, when the query is somewhat general, ambiguous or short, the initial search result is likely to be unreliable. Consequently, pseudo-cluster selection can perform poorly in these situations. While the average number of terms of queries in our relevance judgment set is 2.6, the average number for the queries in the blog distillation judgment set is 1.9. This difference might be critical for pseudo-cluster selection.

On the other hand, the combination of global representation and pseudo-cluster selection significantly outperformed the baselines. In fact, the MAP score is as good as the best reported in the TREC 2007 blog distillation task [32, 94]. While the best run achieved the performance by a novel query expansion technique, our method uses a simple post-processing of query likelihoods, which does not require any other information but a posting index. That is, this approach is very effective for the blog distillation task as well as for the blog site search.

Penalty factors by random sampling were still effective but not as much as that they showed on our dataset. In particular, the method using recent postings as a penalty factor, which showed the best performance on our dataset, was worse than the method using random postings. This presents that the current blog distillation task does not pursue recency and weighing on recent postings is an inappropriate strategy for the blog distillation task. However, topics addressed by blogs often change. Considering that the blog distillation task is a filtering task for future postings, the importance of recent topics of blogs might be improperly overlooked in the judgment process for the blog distillation task.

Although the method using the global representation score as a penalty factor outperformed the random sampling approaches, the differences are not statistically significant. Considering the practical advantage of the random sampling methods

which do not require additional indexes, the methods should be taken into account for the blog distillation task.

4.6 Conclusion

In this chapter, we addressed how to exploit hierarchical structures in social application via the blog site search task. Based on this goal, we introduced two contexts which can be extracted from hierarchical structures and various blog site representation techniques based on these contexts. Furthermore, we classified the types of blog sites and claimed that an appropriate penalty factor derived from a global context reflecting the diversity of the topics of each blog site is required. Our experiments showed that the score of the global representation method can be a good candidate for this factor. Our experiments also demonstrated that combinations of global contexts and local contexts, i.e., pseudo-cluster selection combined with a global representation penalty outperformed the other methods, both on our data and for the TREC Blog Distillation task.

CHAPTER 5

CONVERSATIONAL STRUCTURES AND ONLINE COMMUNITY SEARCH

Although conversational structures exist in various social applications, their most crucial role can be found in online communities such as forums and newsgroups. Specifically, a conversational structure appearing in a thread, which is often called “thread structure”, is defined by reply relations. A general web page can be seen as a monologue where the utterance is a one-way communication by the page’s creator. A community-based question answering (CQA) “document”, which consists of a question and the replies, is a special case of a dialogue where the number of utterances per participant is typically limited to one. In contrast, many-to-many conversations occur frequently in threads. This is an advantageous feature that encourages in-depth discussion, compared to general web pages or CQA services. Also, since every member can correct and update information in a thread via such conversations, information in forums is more reliable and often richer in terms of representing different perspectives. Of course, retrieval techniques can benefit these rich structures.

In this chapter, we focus on search using conversational structures by addressing online community search tasks. Specifically, we set two goals considering conversational structures of threads for online community search. The first goal is to discover and annotate thread structures which are based on interactions between community members. In some community sites, thread structure is explicitly annotated. In many others, however, the annotation is missing or inaccurate. We introduce and evaluate techniques that learn to annotate thread structure based on various features that reflect aspects of interactions between postings.

The second goal is to improve retrieval performance for online community search by exploiting the thread structure. We introduce retrieval models that incorporate thread structures and investigate the effects of threads on retrieval performance. The new retrieval techniques are evaluated using test collections created from two online forums and an email archive.

5.1 Discovery of Thread Structure

A thread is started on some subject by an initiator and grows as people discuss the subject. Since the first posting of the initiator is usually displayed on the top of a thread, we call it the top posting. The top posting can be any utterance which invites interaction with people, e.g., a question, a suggestion, a claim, or a complaint. If they are interested in the subject of the top posting, people post their opinions in reply postings. The reply postings can be any reaction to the top posting, e.g., an answer, agreement, disagreement, advice, or sometimes an additional question. Often, a reply posting may elicit its own replies. This is a typical phenomenon of a discussion in a thread. Because not all reply postings reply to the top posting, many branches (sub-threads) of discussion appear in a thread, and a thread ends up with a tree-shaped structure. We refer to this as a thread structure. Figure 5.1 shows an example of a manually annotated structure of a thread, where a node represents a posting, an arc represents a reply relation between two postings, and each number is a chronological order. That is, the child posting with the outgoing arc replies to the parent posting with the incoming arc.

Not all online communities, however, handle threads in the same manner. There are generally two ways that online communities maintain or display threads: flat-view and threaded-view. Flat-view systems, as their name implies, flatten structures of threads and show users all postings in a thread in chronological order. On the other hand, threaded-view systems allow a user to choose a preceding posting to reply to,

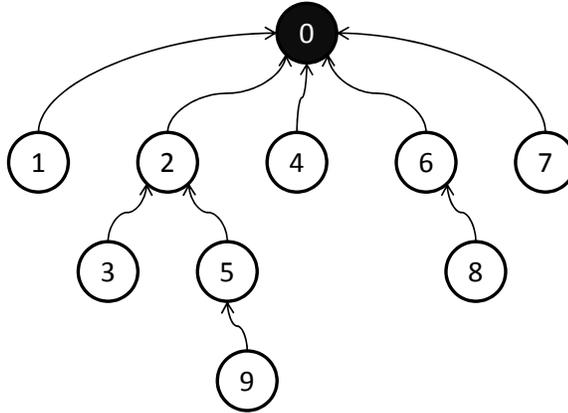


Figure 5.1. Example of a thread structure

and display postings in structured views. Figure 5.2 shows a user-view example of a threaded-view system.

The flat-view looks natural because it resembles aspects of real conversations. Further, the flat view is sometimes more readable than the threaded-view. In particular, if a thread is very long, then it may be difficult for people to grasp all contents of threads in complicated structured views.

On the other hand, if we want to know how discussions flow or how postings interact, the threaded view is more helpful. In particular, if a thread is large, the thread may address many topics, each slightly different to each other. Then we can split the thread into smaller topical units according to the branches of the thread structure. An atomic topical unit such as a passage is known to be useful for information retrieval. Additionally, the threaded view appears to be suitable for social media analysis tasks such as expert finding.

Currently, flat-view online community pages are still much more prevalent although some online communities have emerged that use the threaded view, such as

<p>clig</p> <p>Posts: 3 From: stafford Registered: Aug 19, 2010</p>	<p>Re: bluetooth update</p> <p>New! Posted: Sep 9, 2010 6:39 PM Reply Email</p> <hr/> <p>having updated to 4.1 .apple have still not sorted bluetooth problem.why do they not listen to us</p> <p>dell inspiron1 laptop Windows 7</p>
<p>dbx2spc</p> <p>Posts: 3 Registered: Sep 9, 2010</p>	<p>Re: bluetooth update</p> <p>New! Posted: Sep 9, 2010 7:10 PM in response to: clig Reply Email</p> <hr/> <p>Same here, I can pair it with my Pioneer DEH-P7000BT but once I turn off the car then restart only the phone works. I have to manually pair it again to get the bluetooth audio to work every time. APPLE PLEASE FIX THIS IT IS YOUR ISSUE NOT PIONEERS OR ANY OTHERS.</p> <p>DEH-P7000BT iOS 4</p>
<p>jacksTLOS</p> <p>Posts: 10 From: Mesa, AZ Registered: Sep 2, 2010</p>	<p>Re: bluetooth update</p> <p>New! Posted: Sep 10, 2010 1:01 AM in response to: dbx2spc Reply Email</p> <hr/> <p>Reset your settings:</p> <p>Settings- general- reset- reset all settings</p> <p>This should fix your problems</p> <p>Vaio Windows 7</p>
<p>dbx2spc</p> <p>Posts: 3 Registered: Sep 9, 2010</p>	<p>Re: bluetooth update</p> <p>New! Posted: Sep 10, 2010 10:08 AM in response to: jacksTLOS Reply Email</p> <hr/> <p>Sorry but resetting does not do a **** bit of good.</p> <p>iOS 4</p>
<p>Mike Johnson12</p> <p></p> <p>Posts: 4,426 From: Asia Registered: Jun 2, 2005</p>	<p>Re: bluetooth update</p> <p>New! Posted: Sep 10, 2010 1:11 AM in response to: dbx2spc Reply Email</p> <hr/> <p>Complaining in the Forums will not get any response from Apple.</p> <p>Use the Feed Back links - http://www.apple.com/feedback/</p> <p>MJ</p> <p>MBP 2.66/4/500 - iTunes 10 - TV - iPhone 3GS 32 gig MacOS X (10.6.4) iPod Touch 32 - iPod Photo 60 gig - iPod 2nd gen 20 gig - Blue Shuffle</p>

Figure 5.2. Example of the threaded-view. An indentation indicates a reply relation.

Slashdot¹ and Apple Discussion². One reason for this is that many online forums use popular publishing software such as phpBB³ and vBulletin⁴. Most of these tools either don't support a threaded view or don't provide it as a default. Considering the small number of online communities which support threaded views, we believe that techniques for converting flat-view threads to threaded-view threads are needed for online community search, data mining, and social media analysis. We refer to this conversion as discovery of thread structures.

For simplicity and clarity, we make a number of assumptions about the thread structure discovery task. First, we assume that a thread structure is shaped like a rooted tree in which the top posting is a root, each child posting has only one parent posting, and no node is isolated. Although there may be some cases which violate this assumption, such as answering questions from two postings, these cases are not frequent and, furthermore, most threaded-view systems make the same assumption. The second assumption is that we can find a parent-child (reply) relation considering only pairs of postings. In other words, a reply relation between two postings is independent of their grandparents and grandchildren. Lastly, we assume that a chronological order of postings in a thread is known so that we can consider only the preceding postings of a child posting as candidate parent postings.

These assumptions significantly reduce the complexity of thread structure discovery. Under the first assumption, there are only $N - 1$ reply relations, where N is the number of postings. Further, when we are given a child posting, we can find a reply relation by picking a most likely parent posting from among all preceding postings. Under the second assumption, a greedy approach is the optimal approach to find a

¹<http://slashdot.org/>

²<http://discussions.apple.com/>

³<http://www.phpbb.com/>

⁴<http://www.vbulletin.com/>

Input: P
Output: A
<pre> 1: $A \leftarrow \mathbf{0}$ 2: for $i \leftarrow 1$ to $P - 1$ 3: $L \leftarrow \mathbf{0}$ 4: for $j \leftarrow 0$ to $i-1$ 5: $L[j] \leftarrow \text{compute_reply_likelihood}(P[i], P[j])$ 6: end 7: $A[i] \leftarrow \text{argmax}_j L[j]$ 8: end 9: return A </pre>

Figure 5.3. Algorithm for finding all reply relations in a thread. P is a list of postings in chronological order. A is a list of the indices of corresponding parents of postings.

thread structure. That is, if we can find a correct parent posting for each posting, then we can build a correct thread structure. Finally, the third assumption simplifies the problem because we know which postings precede others.

Constructing a thread structure given reply relations is trivial; thus, finally, our problem is reduced to finding reply relations. Our algorithm for finding all reply relations in a thread is described as shown in Figure 5.3. This requires only $O(N^2)$ pairwise comparisons.

In the next section, we introduce the features used for reply relation detection and a process for learning the `compute_reply_likelihood()` function in Figure 5.3. Finally, we evaluate the performance of our algorithm using experimental results.

5.1.1 Intrinsic Features

A straightforward method that we can use to determine a reply relation between two postings is to directly look at the contents of the postings. If two postings address a similar topic, then the postings are likely to have a reply relation. Further, we can frequently observe that a child posting quotes or reuses text from the parent posting. That is, word or phrase overlap can be evidence of a reply relation between postings.

We use text similarity as a feature in order to address both topical similarity and text overlap. There are numerous measures of text similarity. Among them, we use the *idf*-weighted cosine similarity. Cosine similarity is not only simple but also works well for many IR tasks. Further, since a posting is usually short and *tf* does not often function as more than an indicator of a term occurrence, it is necessary to use *idf* to weight topical terms. The following variation [16] of the *idf*-weighted cosine similarity is used.

$$\text{sim}(\vec{p}_1|\vec{p}_2) = \frac{\sum_{k=1}^m d_k \cdot q_k}{\sqrt{\sum_{k=1}^m d_k^2} \sqrt{\sum_{k=1}^m q_k^2}}$$

$$d_k = 1 + \log tf_{1k}, \quad q_k = (1 + \log tf_{2k}) \log \frac{D + 1}{df_k}$$

where \vec{p}_1 and \vec{p}_2 are word vectors of a parent candidate and a child posting respectively, m is the size of vocabulary, tf is a term frequency, df is a document (posting) frequency, and D is the total number of postings in the collection. A drawback of the *idf*-weighted cosine similarity is that it is non-symmetrical. However, our task is to find the most likely parent posting among the preceding postings of a posting, similarly to traditional information retrieval tasks. In this setting, we do not need to consider reverse relations of the parent posting and the reply posting; thus, symmetry is not necessary.

Our thread structure discovery technique did not empirically show large variance over different similarity measures. Therefore, the other measures can be used if required. Nevertheless, the variation of *idf*-weighted cosine similarity worked best in our experiments; we report only the results using the measure in this chapter.

Now we consider which part of a posting the similarity measure is applied to. There is also the issue of how term vectors are constructed.

- **Quotation vs. Original Content**

Many online community systems support an option to quote text from the

preceding posting when a posting is uploaded. Such systems provide split views of the quotation and the original content. For example, some systems split views using special tags whereas others use some special characters such as ‘)’ in the beginning of the quoted line. In such systems, we can easily determine which text is quoted.

Once we obtain the quotation and the original content separately, we can consider various combinations for similarity measurements. First, we can measure the similarity between the original content of a parent candidate and the original content of a child posting. This similarity is to measure topical similarity between the postings. Second, similarity between the original content of a parent candidate and a quotation of a child posting can be considered. This similarity shows how text is reused between the postings. Last, we can measure similarity between the full texts of postings without separating the quotation from the original content.

- **Unigram vs. n-gram**

We can construct a term vector of a posting with unigrams or n-grams. The fact that two postings share the same phrases or compound words rather than single words can be strong evidence for both text reuse and topical similarity. Therefore, if term vectors are composed of n-grams, we may expect more accurate discovery results. However, most n-gram terms are scarce and the vector space would be sparse. Accordingly, using n-grams can be unreliable in some cases. We will empirically investigate how different constructions of term vectors have effects on discovery results.

5.1.2 Extrinsic Features

A posting is an utterance in an informal dialogue rather than a speech or formal writing. While a few online communities such as technical email archives or political

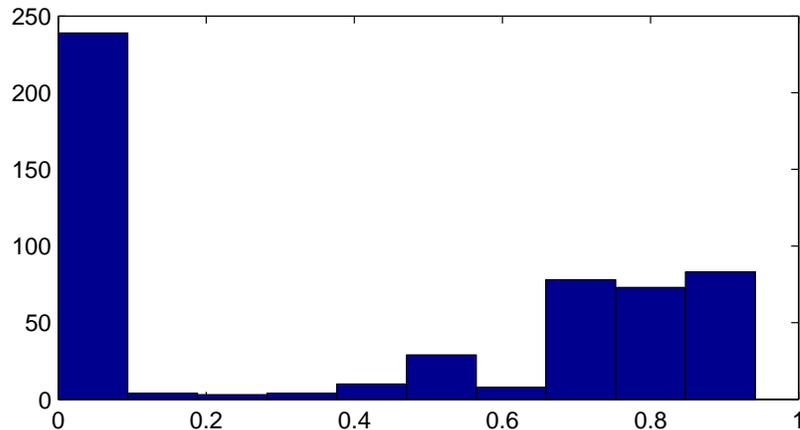


Figure 5.4. Histogram of normalized location indices

discussion forum are formal, many online communities such as game forums, social forums, or travel forums are generally informal. That is, in many cases, a posting tends to be short and “instant”. Therefore, similarity features are not often enough to capture relations between postings because of sparse word distributions. For example, a posting asks a question, “What is the best authentic Mexican food?”, and the next posting says “Taco!” Although the two postings clearly have a question-answer relation through the context, the relation cannot be discovered with similarity features only. Thus, we need to use features which can describe context as well as content. Here we introduce several of these extrinsic features.

- **Location Prior**

Most online community systems provide a view of postings in a thread in chronological order. We can assume that a relation between postings is inferred from the locations of the postings in the chronological time frame. For example, the top posting in a thread has 0 as its location index, and the n th posting in chronological order has $n - 1$ as its location index. If the thread actually has a chronological structure like a dialogue by two individuals, then each posting replies to the immediate preceding posting. In other words, a posting with location index i replies to a posting with location index $i - 1$. On the other hand,

if a thread has a structure in which the top posting asks a question and the others answer the question, then the parent posting of every posting is the top posting with location index 0.

We want to predict where a parent posting is located when the location of a child posting is given. Formally, we want to estimate $Pr(i_1|i_2)$, that is, the likelihood that a posting with location index i_1 is a parent posting of a child posting with location index i_2 . We can directly extract an empirical distribution of the likelihood from annotated thread structures. However, because the amount of annotated data is not enough, each conditional distribution given the location index of each child posting may be inaccurately estimated by sparse data. As a solution, we normalize location indices by the location index of a child posting, i.e., i_1/i_2 and i_2/i_2 . We refer to the normalized value as a normalized location index. We then estimate the likelihood using normalized location indices instead of real location indices. For example, if the original location indices i_1 and i_2 are 3 and 7, then the normalized location indices are $3/7 = 0.43$ and $7/7 = 1$ respectively. Therefore, all normalized location indices fall into $[0, 1]$.

Figure 5.4 shows a histogram of normalized location indices of related posting pairs in the Cancun dataset (See Chapter 5.1.4 for a detailed description of the dataset). As we see, there are two peaks in the histogram. A higher peak is located around 0 and a lower peak is located around 0.8. The former shows how many relations are biased toward the top posting and the latter shows how many relations are biased toward the immediate preceding posting. Relations with the immediate preceding posting can be interpreted as chronological ordering. These two peaks commonly appear in all collections that we use.

We consider the distribution as a Gaussian Mixture which consists of two Gaussian distributions and estimate the mixture by the Expectation-Minimization [12]. Given the estimated distribution and location indices of two postings, we

can compute the likelihood of a relation between the postings as follows:

$$Pr(i_1|i_2) = F_L\left(\frac{i_1 + 1}{i_2}\right) - F_L\left(\frac{i_1}{i_2}\right)$$

where F_L is a cumulative distribution function (cdf) of the estimated distribution. We refer to this likelihood as a location prior and use it as an extrinsic feature.

Note that this estimated prior worked better in preliminary experiments although a location itself can be considered as a feature. In fact, as shown in Figure 5.4, a location cannot be considered as a monotonic feature.

- **Time Gap**

A difference between posting times of two postings can be evidence of a relation between the postings. If a posting is created 10 months after the other posting was posted, then the chance that the postings have any relation is probably small. Conversely, if two postings are sequentially posted with a small time gap, then the chance of a relation increases.

Since the posting time difference has a wide value range, we need to normalize the difference as follows:

$$gap(t_1|t_2) = \frac{t_2 - t_1}{t_2 - t_0}$$

where t_0 , t_1 , and t_2 are the posting times of the top posting, a parent candidate posting, and a child posting. The differences are computed in second. We refer to this normalized value as a time gap.

- **Same Author**

Assuming that turn-taking between speakers happens in a thread, the fact that

two postings are written by the same author usually can be used as negative evidence of a relation. We use an indicator of the same author relationship as a feature, that is, 1 if the authors of two postings are the same, 0 otherwise.

- **Author Reference**

In flat-view systems, it is not easy to tell which posting a posting is replying to. Accordingly, users often refer to the author of the specific posting by writing the name or ID of the author in order to express an intention to reply to a specific posting. We call this behavior an author reference. Existence of an author reference between two postings can be explicit evidence of a relation. We use an indicator of an author reference as a feature, that is, 1 if there is an author reference, 0 otherwise.

- **Inferred Turn-taking**

This feature is derived from a same author relation and an author reference relation. Let posting A , B and C be posted in this order in a thread. If posting A and B have an author reference and posting A and C have a same author relation, then we can infer that posting C replies to posting B when assuming turn-taking with $A \rightarrow B \rightarrow C$. We call the inferred relation between posting B and C an inferred turn-take and express it as an indicator, that is, 1 if there is an inferred turn-take, 0 otherwise. Note that this does not break our second assumption about independence of grandparents because we do not use a relation but a feature extracted from preceding postings.

5.1.3 Learning

We consider the thread structure discovery task as a ranking task. That is, if each child posting is considered as a query, parent candidate postings are considered as documents to be retrieved. Since a posting has only one parent posting, we have only one relevant document for each query. Although our task can be seen as a classification

task or regression task, the strength of a relation between two objects is relative to other relations. Therefore, it sounds feasible to model relative preferences rather than an absolute decision boundary. Indeed, we conducted preliminary experiments using a linear regression algorithm, but ranking algorithms consistently showed better performance.

Since we have several heterogeneous features, it seems inappropriate to use traditional information retrieval techniques. Instead, we use the RankSVM algorithm [57] because it is known to address such settings well. RankSVM learns a ranking function based on pairwise labels by solving an optimization problem as follows:

$$\begin{aligned} \min_{\vec{w}} M(\vec{w}) &= \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i,j} \xi_{ij} \\ \text{subject to } \langle \vec{w}, x_{iR} - x_{ij} \rangle &\geq 1 - \xi_{ij} \\ \forall_i \forall_j \xi_{i,j} &\geq 0 \end{aligned}$$

where x_{iR} is a feature vector extracted from a relation between child posting p_i and its parent posting, x_{ij} is a feature vector extracted from a relation from p_i and non-parent posting p_j , w is a weight vector of a ranking function. We use a linear kernel for RankSVM. Finally, the learned ranking function is the `compute_reply_likelihood()` in Figure 5.3.

5.1.4 Collections

We use three online community collections in order to evaluate techniques for thread structure discovery. Two of them are online forums. The other is an email archive. The detailed statistics of each collection are presented in Table 5.1.

Table 5.1. Statistics of collections

	WOW	Cancun	W3C
#threads	16,274	58,150	72,214
avg. # postings per thread	84.4	9.1	2.1
avg. posting length (in words)	57.3	67.0	249.6
size (in Gigabytes)	14.0	7.0	3.4

- **World of Warcraft (WOW) forum**

We crawled the general discussion forum⁵ of the World of Warcraft (WOW), a popular online game. The collection contains threads created from August 1, 2006 to April 1, 2008. Among our three collections, the WOW collection is the most casual online community. Most users of online game forums are in the youth demographic. Many postings are not only short but also poorly composed. We can frequently observe broken English, typos and abbreviations. Furthermore, people tend to write postings without serious thought, which often results in long threads as shown in Table 1.

We randomly picked 60 threads which contain at least 5 postings. We split them into 2 sets of 40 threads with overlap of 20 threads, and assigned the sets to two annotators. An annotator tagged all reply relations between postings in each thread in the given set, i.e., 1 if a pair of postings is a reply relation, 0 otherwise. To merge the annotations for the overlap threads, we took 10 threads from each annotator, e.g., odd numbered threads from annotator 1 and even numbered threads from annotator 2. Cohen’s kappa, the inter-annotator agreement computed with the annotations of the overlap threads, was 0.88.

We can extract all the features that we introduced earlier from the WOW collection. In particular, the WOW forum displays the quotation and the original

⁵<http://forums.worldofwarcraft.com/board.html?forumId=10001>

content differently using HTML tags. Therefore, we could extract quotations and original contents separately using simple rules.

- **Cancun forum**

We crawled the Cancun forum⁶ of tripadvisor.com, a popular travel guide site. The Cancun collection contains threads accumulated for about 4 years from September 7, 2004 to November 23, 2008. The Cancun forum is somewhat more formal than the WOW forum. Postings are relatively well written, and the average length of postings is longer than the WOW forum.

We annotated structures of 60 threads through the same process as the WOW forum. Cohen’s kappa of the Cancun forum annotations was 0.86.

A major difference of the Cancun forum to the WOW forum is that the Cancun form does not systemically support quotation. Therefore, we cannot extract quotations and original contents separately.

- **W3C email archive**

We also used the ‘lists’ sub-collection of the W3C collection from the email discussion search task of the TREC enterprise track [124]. The collection was crawled from the mailing list⁷ of the World Wide Web Consortium (W3C). Email archives or newsgroups are old-style online communities but are still active in technical areas. The W3C collection is the most formal of our collections. Most participants are scholars or experts in the field and most postings are written in a polite tone. As you see in Table 1, the average length of a posting is much longer than the other collections.

⁶<http://www.tripadvisor.in/ShowForum-g150807-i8>

⁷<http://lists.w3c.org/>

The W3C collection provides thread structures in the ‘`thread.html`’ file in each group archive. However, many of these thread structures are wrong. We frequently find cases where an earlier email replies to a later email. This is because the ‘`msg-id`’ and ‘`inreply-to`’ tags in email headers are often lost. A thread of emails is usually constructed by matching tags. If they are missing, then email archive tools infer threads using heuristics such as title matching. Such inferences are often inaccurate.

To build an annotation set for thread structure discovery, we refined the thread structures by picking threads only composed of emails whose ‘`inreply-to`’ tag matches a ‘`msg-id`’ tag of any other posting in the same thread. Finally, in this set, we obtained 1635 threads which contain at least 3 emails.

All features that we introduced earlier are available in the W3C collection. Since quoted text begins with some special characters such as ‘`’`’, we can easily divide each message into the quotation and the original content. We removed all lines which start with multiple special characters because they are a part of replies to replies which we do not consider in our thread structure discovery task.

Note that we refer to an email as a posting in other sections of this chapter for consistency.

5.1.5 Experiments

We conducted experiments for thread structure discovery on each collection. To investigate the effectiveness of features, we tested various combinations.

We compute accuracy to evaluate the performance of each combination of features as follows.

$$accuracy = \frac{|\{\text{reply relations}\} \cap \{\text{detected relations}\}|}{|\{\text{reply relations}\}|}$$

Accuracy is computed for each thread, and the final evaluation measure is the average of accuracy scores. Note that, in this setting, this metric is the same as recall or precision because they have the same denominator (i.e., the number of postings in a thread - 1). Also, we can employ other information retrieval evaluation metrics such as mean reciprocal rank (MRR) because our task is considered as a ranking task. However, the fact that a true reply relation is highly ranked by our algorithm as long as the relation is not located at rank 1, does not affect the discovered thread structure. This is a difference from other retrieval tasks such as ad hoc retrieval where a ranked list is generally provided to users. Accordingly, we do not consider such metrics for evaluation.

For the WOW and Cancun collections, because the annotated data is small, we performed 10-fold cross validation for evaluation, that is, we used 54 threads per partition as training data. On the other hand, since the W3C collection has enough data for training, i.e. 1,635 threads, we used 1,535 threads as training data and 100 threads as test data.

For intrinsic feature extraction, only the title and body text in each posting were used. The text was pre-processed by the Porter stemmer [101] and stopword removal.

5.1.6 Results and Discussion

Table 5.2, 5.3 and 5.4 show the experimental results for the three collections. In the tables, each row corresponds to an intrinsic feature and each column corresponds to an extrinsic feature.

In the WOW collection, the similarity of quotations is more helpful than topical similarity of original contents. However, we can see a performance gain from using both of them. Unigram and n-gram do not show significant differences in performance. Among the extrinsic features, the location prior and the time gap are the most helpful features. When using either of them, we see improvements of at least 20%. The best

Table 5.2. Thread structure discovery results on the WOW collection. Values are accuracy scores. Each row corresponds to an intrinsic feature: full text (F), original contents (O), quotations (Q), unigram (U) and n-gram (N). Each column corresponds to an extrinsic feature: location prior (LP), time gap (TG), author reference (AR), same author (SA), inferred turn-taking (IT) and all extrinsic features (ALL). Bold values indicate the best score group, i.e., the score is not statistically significantly different from the best score (by the paired randomization test with p -value < 0.05).

	None	LP	TG	AR	SA	IT	All
None		0.5770	0.5867	0.2959	0.2996	0.2890	0.5302
F+U	0.5858	0.7629	0.7223	0.5901	0.6140	0.5858	0.8025
F+N	0.5856	0.7908	0.7249	0.5880	0.6129	0.5856	0.8125
O+U	0.4364	0.5704	0.5103	0.4421	0.4469	0.4374	0.5745
O+N	0.4346	0.5738	0.5131	0.4403	0.4469	0.4356	0.5770
Q+U	0.5791	0.8814	0.8824	0.5824	0.5613	0.5791	0.8698
Q+N	0.5779	0.8809	0.8873	0.5812	0.5672	0.5779	0.8842
O+Q+U	0.6570	0.8922	0.8228	0.6604	0.6534	0.6570	0.8726
O+Q+N	0.6531	0.8851	0.8234	0.6564	0.6502	0.6531	0.8798

Table 5.3. Thread structure discovery results on the Cancun collection

	None	LP	TG	AR	SA	IT	All
None		0.4839	0.4861	0.5104	0.4034	0.4139	0.5630
O+U	0.4697	0.5057	0.5563	0.4922	0.5159	0.4840	0.6165
O+N	0.4656	0.5083	0.5509	0.4862	0.5025	0.4818	0.6279

Table 5.4. Thread structure discovery results on the W3C collection

	None	LP	TG	AR	SA	IT	All
None		0.7149	0.7284	0.7156	0.6520	0.6726	0.7811
F+U	0.8988	0.8785	0.9017	0.8954	0.9210	0.9104	0.9162
F+N	0.9065	0.8996	0.9137	0.9200	0.9336	0.9114	0.9343
O+U	0.6317	0.6973	0.7658	0.7134	0.7397	0.7138	0.8053
O+N	0.6309	0.6966	0.7621	0.7152	0.7380	0.7130	0.8061
Q+U	0.8907	0.9130	0.9078	0.9058	0.8986	0.9066	0.9351
Q+N	0.8907	0.9130	0.9078	0.9058	0.8986	0.9066	0.9343
O+Q+U	0.9067	0.9133	0.9282	0.9354	0.9366	0.9273	0.9533
O+Q+N	0.9170	0.9183	0.9393	0.9295	0.9457	0.9222	0.9617

combinations require at least similarity of quotations as an intrinsic feature and either or both of the location prior and the time gap as an extrinsic feature. The best scores have almost 90% accuracy.

In the Cancun collection, the scores are much worse than those of the WOW collection. This is mainly because the Cancun collection does not have any quotations. On the basis solely of the non-quotation features, the performance in the Cancun collection is similar to or better than the WOW collection. Another difference from the WOW results is that author reference is more effective. We hypothesize that users refer to other postings more frequently in the Cancun collection because they cannot use quotations supported by the forum system. In addition, the location prior and the time gap are also helpful. The best performance is achieved when all features are used.

In the W3C collection, we see very good results even using only the intrinsic features. Quotations, in particular, are very helpful. In emails, not only is text usually long enough, but also the whole text of each mail is almost always quoted by a reply. The high accuracy obtained by the intrinsic features can be explained by these characteristics of email. However, we still observe performance gains from using extrinsic features in addition to intrinsic features.

For baselines for comparison, we can assume specific thread structures. Specifically, two simple structures can be considered. The first is that all postings reply to the top posting. We call this a top-based structure. The second structure is that all postings reply to the immediate preceding postings. We call this a chronological structure.

Another baseline to consider is a graph-based propagation algorithm introduced by Cong et al. [26]. Although the algorithm is used for detecting relevant answer postings for a question posting in a forum thread, their task is similar to ours in that they also seek relations between postings in a thread. The graph-based propagation

Table 5.5. Thread structure discovery accuracy on baselines. Two baselines (the first and second rows) consider specific thread structures, i.e., the top-based structure and the chronological structure. Another baseline (the third row) uses the graph-based propagation algorithm [26].

	WOW	CANCUN	W3C
Top-based	0.5773	0.5202	0.4676
Chronological	0.2713	0.4839	0.7161
Graph-based Propagation	0.3132	0.5315	0.6526

algorithm performs a random walk on a directed graph which encodes inter-posting relations with edge weights computed by:

$$w(p_1 \rightarrow p_2) = \frac{1}{1 + KL(p_1||p_2)} + \lambda_1 \frac{1}{dist(q, p_2)} + \lambda_2 authority(p_2)$$

where q is a query posting, p_1 and p_2 any two candidate postings in the same thread, $KL(p_1||p_2)$ is the Kullback-Leibler divergence of language models of p_1 and p_2 , and $dist(q, p_2)$ is the locational distance between q and p_2 . *authority* of a posting is computed by normalizing ($\#reply^2/\#start$) where $\#reply$ is the number of replies by the author of p_2 and $\#start$ is the number of threads initiated by the author. λ_1 and λ_2 are linear combination parameters which were set to the same values as reported in [26]. From this formula, we can know that this algorithm tries to incorporate similarity, locational information and authorship information of postings into a graph. Postings are ranked by the stationary distribution obtained by a random walk on this graph; then, the relation between the first ranked posting and the question posting is predicted as a reply relation.

Table 5.5 shows the results of thread structure discovery using the baselines. Interestingly, each collection shows a different aspect. The WOW forum is biased toward the top-based structure. This shows that people tend to read only the top posting and reply to it because a thread in the WOW forum is often very long as shown in Table 1. Conversely, the W3C collection is biased toward the chronological structure.

Although the W3C archive is a public community based on a mailing list, the characteristic of the discussions is more private compared to online forums. That is, a discussion is often similar to one-to-one conversation rather than a group discussion even though everyone can listen to it. Since each participant knows all issues in the preceding mails, a new mail naturally tends to be a reply to the immediate preceding mail. In the Cancun forum, the two specific structures are almost equally likely. This shows that the different aspects of the other two online communities are mixed in the Cancun forum.

Comparing the performances of the baselines to ours, our algorithm significantly outperforms discovery based on the specific structures regardless of types of online communities. This presents that threads cannot be assumed to have a simple structure. Also, the graph-based propagation algorithm shows significantly worse performance than the best performance of our algorithm. This is because the graph-based propagation algorithm tries to identify a relevant posting (which is often created by an authoritative author or informative) to a query posting rather than a real parent posting of a child posting in a thread structure. For example, a highly relevant posting may appear after a long discussion involving a number of postings following a query posting. The graph-based propagation algorithm picks up the posting even when it is not a direct reply to the query posting, whereas we would like to reconstruct all contexts via direct reply relations.

One question is what features should be used in practice. The answer is simple: If all features are available, use them all. For the Cancun and the W3C collection, the best accuracy is gained when using all features. For the WOW collection, although using all features is not the best, the difference from the best performance is not statistically significant. The most effective intrinsic feature is the similarity of quotations, and there is no notable difference between unigram and n-gram. Therefore, if resources are limited and quotations exist, the best approach for intrinsic features

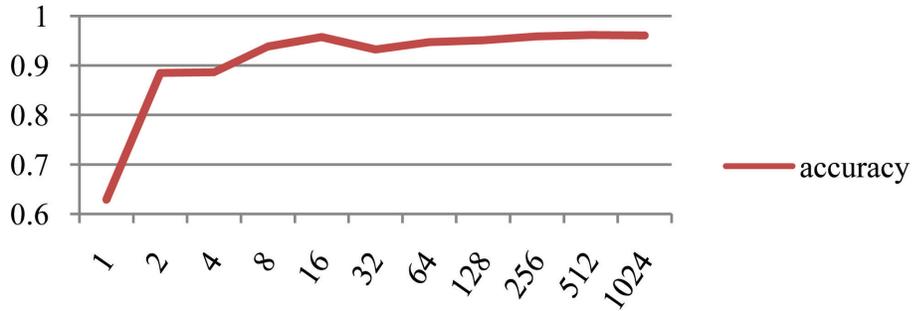


Figure 5.5. Learning curve on the W3C collection. The change of accuracy on test sets (y -axis) depending on the number of threads in the training set (x -axis) is plotted.

is to compute the similarity of only quotations using unigram. For extrinsic features, the location prior and the time gap are almost always effective. The authorship-based features, i.e., the same author, the author reference, and the inferred turn-taking, are shown to be effective only in the formal community such as the W3C where authors' real names are known. In many informal communities such as the WOW and the Cancun, only user IDs are public. Because user IDs are often combinations of alphabets and numbers that the others except the owner cannot understand, in such communities, references do not frequently occur, and we cannot easily recognize the reference even when there is. Accordingly, the effect of the authorship-based features is limited.

There is also the question of how much training data is required to achieve good accuracy. Since the W3C collection has sufficient training data, we plot a learning curve according to the amount of training data as shown in Figure 5.5. We can see that the curve becomes stable from around 50~60 threads. Although this may vary between collections, it provides some support for the size of training data used on the other collections (i.e., 54 threads).

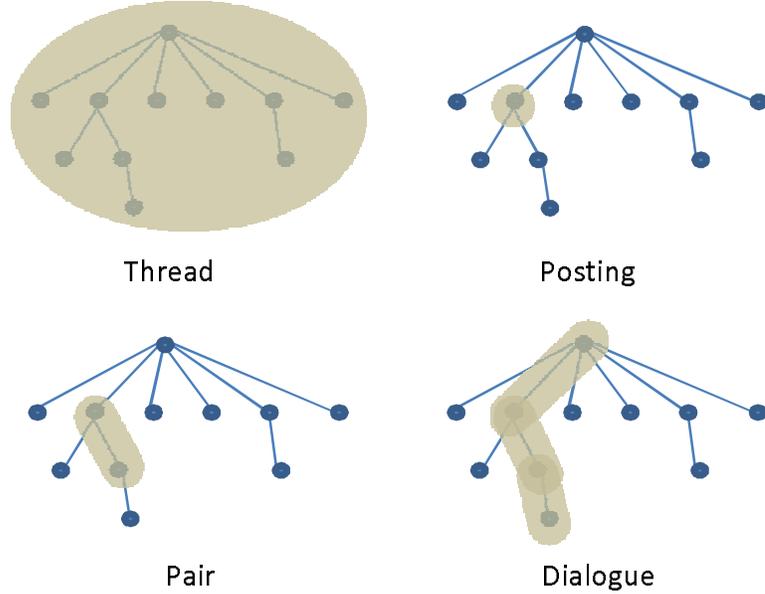


Figure 5.6. Contexts in a thread structure

5.2 Multiple Context-based Retrieval

In this section, we introduce approaches to improve retrieval performance using thread structures discovered by the algorithms introduced in Chapter 5.1.

5.2.1 Context Extraction based on Thread Structure

A document is composed of self-contained text units in various levels, e.g., sentences, paragraphs or sections. Similarly, a thread is composed of different self-contained sub-structures. We call a sub-structure a context.

Figure 5.6 presents four contexts. The first context is the coarsest-grained context, i.e., the thread itself. The second context is the finest-grained context, i.e., a posting. While we can use thread contexts to get a general picture about the topic addressed by a thread, we can use posting contexts to get detailed information. The third context is a pair defined by a reply relation. This context is directly extracted from a relation discovered by thread structure discovery algorithms. A pair context contains an interaction between two users. For example, the context may be a question-answer pair. If what we want is an answer to a question, a pair context can be suitable. The

fourth context contains all postings in a path from the root node (top posting) to a leaf node. We refer to this context as a dialogue because by looking at the context we can follow a conversation flow, e.g., how the discussion was started, what issue was discussed, and what the conclusion was.

Note that we can extract thread contexts and posting contexts without regard to the structure of a thread. However, pair contexts and dialogue contexts must be extracted from a thread structure.

5.2.2 Multi-context-based Retrieval

We address two retrieval tasks using multiple contexts: thread search and posting search. Since postings in casual online forums such as WOW or Cancun are usually too short to provide information on their own, people are likely to want to find relevant threads rather than postings. On the other hand, emails (postings) in a technical email archive like the W3C archive are often long enough to deliver information. In that case, a more suitable task is to find relevant emails (postings).

For these two tasks, we introduce retrieval techniques based on a language modeling approach to retrieval [28]. In our work, the query likelihood $Pr(Q|D)$ is estimated under the term independence assumption as follows:

$$Pr(Q|D) = \prod_{q \in Q} ((1 - \lambda)Pr_{ML}(q|D) + \lambda Pr_{ML}(q|C)) \quad (5.1)$$

where q is a query term in query Q , D is a document, C is the collection, λ is a smoothing parameter, and $Pr_{ML}(\cdot)$ is the maximum likelihood estimate, i.e., $Pr_{ML}(w|D) = tf_{w,D}/|D|$. If we use the Dirichlet smoothing [139], then $\lambda = \mu/(\mu + |D|)$ where μ is a Dirichlet smoothing parameter.

5.2.2.1 Thread Search

The simplest approach to thread search is to consider a thread as a document, i.e., the global representation technique introduced in Chapter 4.

$$\Gamma_{GR}(Q, T_i) = Pr(Q|T_i)$$

where Γ is a ranking function and $Pr(Q|T_i)$ is a query likelihood score of query Q for thread T_i .

A drawback of global representation is that relevant local contexts can be dominated by non-relevant contexts. A thread often addresses a broad topic or a mixture of sub-topics, but user queries may be specific. For example, in an online game forum, while a thread addresses “the best weapons”, a user query may be “the best sword for warriors”. Then, global representation may not locate the thread even when highly relevant local contexts for the query are contained in it. For threads as long as those in the WOW collection, this problem can be serious.

To tackle this drawback, we employ more advanced techniques using discovered structures. For example, we can use the geometric representation technique of Chapter 3. More specifically, pseudo-cluster selection based on the geometric representation can be used for this task because a thread can be considered as a collection of local contexts, i.e., postings, pairs or dialogues, as done in Chapter 4. That is, we first retrieve the top N local contexts and aggregate local contexts in the ranked list according to which thread the local context comes from. Each local context group is called a pseudo-cluster. Finally, relevant threads are located according to a geometric mean of scores of the top K local contexts in a pseudo-cluster as follows:

$$\Gamma_{PCS}(Q, T_i) = \left(\prod_{j=1}^K Pr(Q|L_{ij}) \right)^{1/K} \quad (5.2)$$

where $Pr(Q|L_{ij})$ is a query likelihood score based on the language model of local context L_{ij} in thread T_i .

For a pseudo-cluster which contains fewer than K local contexts, we use the upper bound approach suggested in Chapter 4.

$$L_{\min} = \operatorname{argmin}_{L_{ij}} Pr(Q|L_{ij})$$

$$\Gamma_{PCS}(Q, T_i) = \left(Pr(Q|L_{\min})^{K-m} \prod_{j=1}^m Pr(Q|L_{ij}) \right)^{1/K}$$

where m is the number of local contexts in a pseudo-cluster. This technique has been proved effective for thread search based on posting contexts [34].

Pseudo-cluster selection reflects how much relevant information exists locally in a thread whereas global representation reflects the cohesiveness of the thread. Therefore, we consider a weighted-product of the ranking function of global representation and the ranking function of pseudo-cluster selection to improve retrieval performance as follows:

$$\Gamma_{Product}(Q, T_i) = \Gamma_{PCS}(Q, T_i)^{(1-\pi)} \cdot \Gamma_{GR}(Q, T_i)^\pi \quad (5.3)$$

where π is a weight parameter.

5.2.2.2 Posting Search

We retrieve relevant postings using estimated language models for postings. If we have posting contexts only, language models are estimated using smoothing as follows:

$$Pr(w|D) = (1-\lambda_1)Pr_{ML}(w|D) + \lambda_1Pr_{ML}(w|C) \quad (5.4)$$

where D is a posting, C is the collection, and λ_1 is a smoothing parameter.

If we know that the posting belongs to thread T , then we can do two-stage smoothing similarly to cluster-based retrieval [79]. This is also similar to an effective approach for the email discussion search task of the TREC 2006 Enterprise track [98].

$$Pr(w|D) = (1 - \lambda_1)Pr_{ML}(w|D) + \lambda_1((1 - \lambda_2)Pr_{ML}(w|T) + \lambda_2Pr_{ML}(w|C)) \quad (5.5)$$

Further, if we have another context X_z , i.e., a pair context or a dialogue context, then we can add one more smoothing stage. However, in contrast to thread contexts, a posting can belong to multiple pair contexts or dialogue contexts. We compute a geometric mean to combine language models of the contexts as follows:

$$Pr_z(w|D) = (1 - \lambda_1)Pr_{ML}(w|D) + \lambda_1((1 - \lambda_2)Pr_{ML}(w|X_z) + \lambda_2((1 - \lambda_3)Pr_{ML}(w|T) + \lambda_3Pr_{ML}(w|C))) \quad (5.6)$$

$$Pr(w|D) = \left(\prod_{z=1}^Z Pr_z(w|D) \right)^{1/Z} \quad (5.7)$$

where Z is the number of contexts which contain D .

5.2.3 Test Collections

For retrieval experiments, we used the three collections used for thread structure discovery. While two online forums were used for the thread search task, the W3C collection was used for the posting (email) search.

Since the W3C collection has been used for the email discussion search task of the TREC enterprise track, there is a relevance judgment set provided by TREC, which contains 110 queries and 58,436 relevance judgments [124]. Since our posting search task is almost the same as the email discussion search task, we used these relevance judgments to evaluate posting search in the W3C collection. Note that

Table 5.6. Example queries for the WOW collection and the Cancun collection

WOW	the best solo PvP class how to beat warlock recommended quest chains for level 70s
CANCUN	winter weather in Cancun couple only all inclusive hotel Isla Mujeres tour

Table 5.7. Summary of relevance judgments of two forum collections (WOW and CANCUN). The numbers of judged threads and relevant threads are averaged per topic.

	Avg. # of topics	Avg. # of judged threads	Avg. # of relevant threads	Avg. # of highly relevant threads
WOW	30	86.4	5.7	3.6
CANCUN	30	80.0	14.0	22.1

although the judgments were made in multi-grades, the grade reflects whether an email contains pro/con statement rather than the degree of relevance. Therefore, we used the judgments as binary relevance judgments.

On the other hand, we had to make our own relevance judgments for the other two collections. For each collection, we chose 30 popular titles among titles of threads which were created after our crawl and asked two people to manually generate keyword queries from the titles. Table 5.6 shows a few examples of queries for the WOW and the Cancun collection. We created relevance judgment pools using retrieval techniques in this Chapter and linear mixture models. To make the initial runs, we estimated the Dirichlet smoothing parameter using the variance-based unsupervised estimation method (See the details in Appendix B). We made ternary relevance judgments, i.e., 0 for irrelevant threads, 1 for relevant threads, and 2 for highly relevant threads. In total, we made relevance judgments for 2,591 threads for the WOW collection and 2,401 threads for the Cancun collection. A summary of the relevance judgment sets are presented in Table 5.7.

5.2.4 Experiments

We discovered structures for all threads in each collection using the SVM classifier trained with the best feature combinations in Chapter 5.1 and the algorithm in Figure 5.3. Then, we applied multi-context-based retrieval techniques to contexts extracted from the structures. Text was stemmed by the Krovetz stemmer [66], and no stop-words were removed for retrieval experiments. Note that although we used different stemmers for thread structure discovery and retrieval experiments for convenience in implementing each system, this does not mean that a specific stemmer is preferred for each task.

As evaluation metrics, we used normalized discounted cumulative gain at 10 (NDCG@10) and mean average precision (MAP) for thread search with the WOW collection and the Cancun collection. MAP and precision at 10 (P@10) were used for posting search with the W3C collection. In all cases, MAP and P@10 are computed considering a judged document whose grade is equal to or greater than 1 as relevant.

Dirichlet smoothing was used to estimate language models for all experiments. Accordingly, smoothing parameters (λ , λ_1 , λ_2 and λ_3) in Equation 5.1, 5.4, 5.5 and 5.6 are determined by $\mu/(|D| + \mu)$ where μ is a Dirichlet smoothing parameter for each context or smoothing stage. To evaluate performance, we performed 10-fold cross validation. For thread search, the parameters to be tuned are the Dirichlet smoothing parameters for context language models, the number of postings in a pseudo cluster, and the weight parameter for the combination of GR and PCS. For posting search, the Dirichlet smoothing parameters for each smoothing stage were tuned. The parameters were exhaustively searched to maximize NDCG@10 for thread search and MAP for posting search.

5.2.5 Results

Table 5.8 and 5.9 show results of thread search on the WOW collection and the Cancun collection. ‘Thread’ means global representation based on a thread context. ‘Posting’, ‘Pair’ and ‘Dialogue’ mean pseudo-cluster selection based on each context. ‘+ Thread’ means that a weighted-product of GR and PCS is used. The top three rows in the tables are considered as baselines because they do not need to use structures of threads.

In the WOW collection, techniques based on dialogue contexts show better or at least comparable performance to techniques based on the other contexts. Particularly, when using dialogue contexts and thread contexts together, the best performance is achieved, and the improvements over all baselines are statistically significant. This demonstrates that a performance improvement in thread search can be achieved using thread structures, particularly, dialogue contexts. A weighted-product of GR and PCS shows better performance than solely GR or PCS. The combination of GR and PCS proves to be an effective approach for thread search as well as for blog site search.

In the Cancun collection, similar trends are shown, that is, dialogue context-based search and the combination of GR and PCS consistently present better performance than the others. However, the improvements are not always statistically significant, in contrast to in the WOW collection. This is presumed to be due to the relative inaccuracy of thread structure discovery in the Cancun collection. To justify this assumption, we investigated retrieval performance based on inaccurate thread structures in the WOW collection. To simulate inaccurate discovery, we used unigram similarity in the full text only as a feature (‘F+U’ row, ‘None’ column in Table 2) and applied the best retrieval technique, i.e., ‘Dialogue + Thread’ to contexts extracted from the inaccurate structure. The results are shown in Table 5.10. This performance is not only worse than the performance based on accurate structure discovery but also fails

Table 5.8. Retrieval Performance on the WOW collection (Thread Search). The superscripts α , β and γ indicate statistically significant improvements on each baseline, i.e., ‘Thread’, ‘Posting’, ‘Posting + Thread’, respectively (by the paired randomization test with p -value < 0.05).

	NDCG@10	MAP
Thread	0.4200	0.3705
Posting	0.2966	0.2565
Posting+Thread	0.4519	0.3875
Pair	0.3763 ^{β}	0.2998 ^{β}
Pair+Thread	0.4447 ^{$\alpha\beta$}	0.3885 ^{$\alpha\beta$}
Dialogue	0.4374 ^{β}	0.3599 ^{β}
Dialogue+Thread	0.4823 ^{$\alpha\beta\gamma$}	0.4073 ^{$\alpha\beta\gamma$}

Table 5.9. Retrieval Performance on the Cancun collection (Thread Search)

	NDCG@10	MAP
Thread	0.4612	0.2630
Posting	0.4763	0.2887
Posting+Thread	0.4942	0.2896
Pair	0.4478	0.2413
Pair+Thread	0.4897 ^{α}	0.2857 ^{α}
Dialogue	0.4938 ^{α}	0.2618
Dialogue+Thread	0.5141 ^{$\alpha\beta$}	0.2973 ^{α}

Table 5.10. Retrieval performance of the WOW collection (based on inaccurate thread structure discovery)

	NDCG@10	MAP
Dialogue+Thread	0.4651 ^{$\alpha\beta$}	0.3869 ^{β}

Table 5.11. Retrieval performance on the W3C collection (Posting Search). The superscripts α and β indicate statistically significant improvements on the baselines, i.e., ‘Posting’ and ‘Posting + Thread’, respectively (by the paired randomization test with p -value < 0.05)

	MAP	P@10
Posting	0.2405	0.4404
Posting+Thread	0.2931	0.4945
Posting+Dialogue+Thread	0.3036 $^{\alpha\beta}$	0.5101 $^{\alpha\beta}$
Posting+Pair+Thread	0.3101 $^{\alpha\beta}$	0.5147 $^{\alpha\beta}$

to show significant differences over the baseline ‘Posting+Thread’. This shows that the accuracy of thread structure discovery can be critical in our retrieval framework.

Table 5.11 shows the results of posting search on the W3C collection. Each row represents which contexts are used for smoothing. The one-stage and two-stage smoothing at the top two rows, which use posting contexts and threads contexts only, do not require thread structures. Therefore, we consider them as baselines. For both the pair context and the dialogue context, addition of the thread context for smoothing achieved statistically significant improvements. This shows that contexts based on thread structure are also helpful for posting search.

5.2.6 Comparison with cluster-based language model

A question which raises from the posting search results is whether the improvements really come from thread structures or from other structures implied in the thread structures. For example, since we used similarity among postings as a feature for thread structure discovery, we can guess that similarity structures rather than the thread structures may lead to the improvements. To examine this assumption, we apply a cluster-based language model approach [79], which performs document smoothing with clusters built with similar documents, to the posting search task. In particular, we follow the best performing practice among various techniques introduced in [79]. That is, we made clusters in a query-independent way using the cosine measure for document similarity. To assign documents into a cluster, the agglom-

Table 5.12. Retrieval performance of cluster-based language models on the W3C collection (Posting Search). These results do not show statistically significant differences from the baseline ‘Posting’ in Table 5.11 (by the paired randomization test with p -value < 0.05).

	MAP	P@10
Cluster-based LM	0.2422	0.4541

erative clustering method [31] implemented in the Lemur toolkit ⁸ was used. This resulted in 14,346 clusters for the W3C collections. Using these clusters, we estimate a document language model as follows:

$$Pr(w|D) = (1 - \lambda_1)Pr_{ML}(w|D) + \lambda_1((1 - \lambda_2)Pr_{ML}(w|cl) + \lambda_2Pr_{ML}(w|C))$$

where cl is a cluster which D belongs to. To estimate the cluster language model, a big document is created by concatenating all documents in the cluster. Parameters λ_1 and λ_2 are determined by 10-fold cross validation, as done in the previous experiments.

Table 5.12 shows the posting search results by this model. The results fail to show any significant improvement even on the simplest baseline (‘Posting’ in Table 5.11) which does not use thread structures. This suggests that simple similarity structures without considering thread structures are not helpful for posting search.

5.3 Conclusions

In this chapter, we investigated whether search for community sites such as forums could be improved using conversational structures or thread structures. We defined a thread structure discovery task and introduced various intrinsic and extrinsic features, and algorithms for this task. Our results show that threads can often be accurately identified using our approach. We then introduced retrieval methods based on contexts extracted from the thread structures. We showed that combinations of

⁸<http://www.lemurproject.org/lemur.php>

multiple thread contexts can achieve significant retrieval effectiveness improvements over strong baselines.

CHAPTER 6

SOCIAL STRUCTURES AND EXPERT FINDING

Social structures imply information about relationships among people. In some social applications such as social networking tools, these relationships explicitly exist because making or maintaining connections among users is one of the most important features in such tools. However, in many social applications which can be used as information sources, these relationships are not explicit. For example, online communities such as forums often provide only user profiles. To exploit social structures in these applications, we need to discover implicit social structures by defining the structures so that relationships among users can be revealed.

We define a social structure by authorship or relationships among users appearing in discussions in these applications. When simply considering only authorship, each user can be represented by a set of documents composed by the user. That is, a user can be seen as a topic distribution. Then, social structures are defined by relationships between these topic distributions of different users. For example, if the topic distributions of two users are similar, we can infer that they have similar interests. In addition to authorship, we can also consider relationships among documents composed by users. For example, if a user frequently posts a reply to a specific person's postings, we may infer a certain relationship between them, e.g., friendship. Also, if a user regularly posts answers to question postings related to a specific topic, we may assume that she is an expert in the topic. Indeed, conversational structures discussed in Chapter 5 imply these relationships among documents.

In this chapter, to present how these social structures can be exploited, we focus on finding an influential social role - an expert in online communities. Although members in a community can share their opinions with others without any discrimination, their expertise, in fact, varies greatly. Expert finding is to identify experts in a specific topic and accordingly distinguishes experts from other members. Expert identification in online communities is important for the following two reasons. First, online communities can be viewed as knowledge databases where knowledge is accumulated by interactions among the members. That is, we read articles in online communities to obtain information about specific topics. If we find articles written by experts, we tend to have more confidence in the contents. On the other hand, in terms of communication dynamics, online communities are spaces where even non-experts can communicate with experts. In the real world, communicating with experts is not only difficult but also expensive. However, we can communicate relatively easily and reliably with experts in online communities once we have identified them.

Therefore, we address expert finding tasks. In particular, we introduce graph-based algorithms integrating various structures and evaluate these techniques on an email archive and a forum.

6.1 Graph-based Expert Finding Techniques

An effective approach for expert finding is a two-step approach. That is, we first retrieve a topically relevant document subset and then find experts using the subset [20, 116, 124]. To find the topical subset, we introduce two expertise graph construction methods: posting-based and thread-based graph construction. In fact, the methods can be thought to be based on local contexts and global contexts of hierarchical structures, respectively. Also, we present an approach considering conversational structures as an enhancement for the thread-based method. For finding

experts in the expertise graph, we suggest a variation of a random walk algorithm which analyzes the graphs to rank experts.

6.1.1 Posting-based Graph Construction

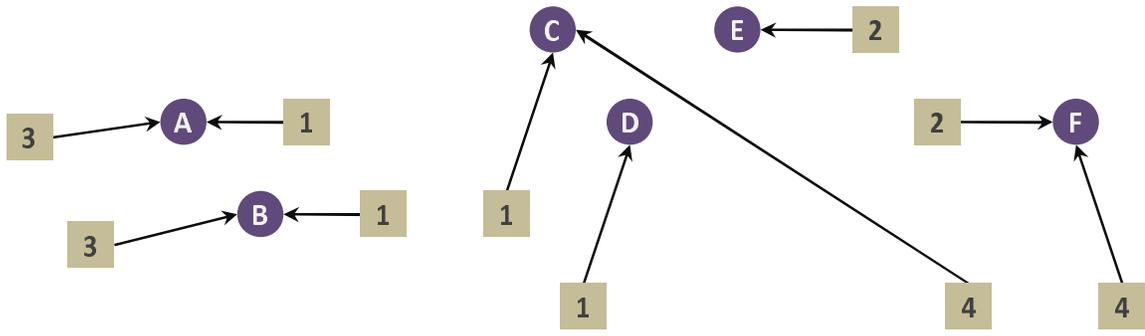
In online communities such as forums, a posting is an atomic topical unit used to communicate with community members. A set of relevant postings can be considered as a relevant subset for expert finding in that a posting usually address a topic and is created by only one person. We assume that we can find experts by analyzing authorship of relevant postings.

To retrieve a set of relevant postings to a given topic, we rank postings by query likelihood scores computed based on Dirichlet smoothed unigram language models of postings. Once we have a ranked list by the query likelihood, we can build a graph using top N postings. First, we make document nodes with the postings. Next, we make candidate expert nodes with unique authors of the postings. Finally, we make directed edges from document nodes to candidate nodes so that each candidate is reachable only from the postings written by the candidate. Figure 6.1(a) presents a posting-based graph example. As you see, each candidate and its postings make a connected graph.

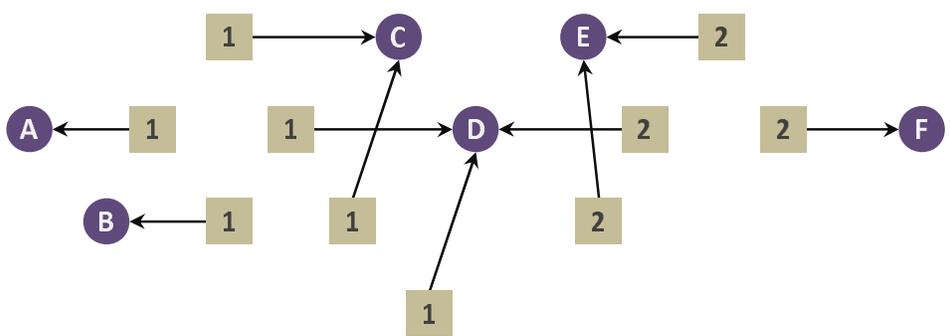
6.1.2 Thread-based Graph Construction

Threads often give better understanding about a topic by contexts or conversational flows than postings, as seen in Chapter 5. Accordingly, we may consider a set of relevant threads as a subset for expert finding.

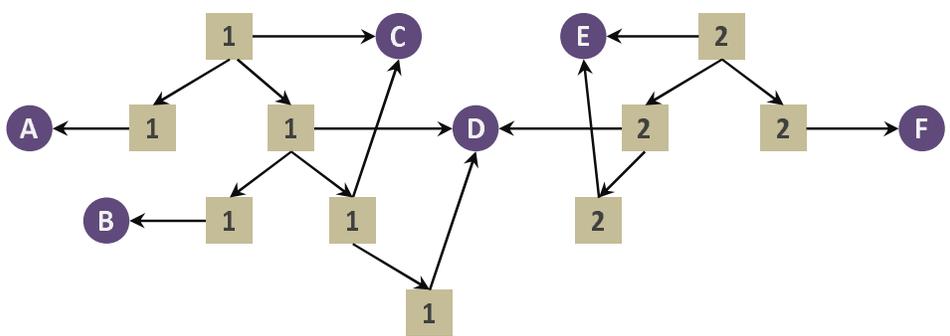
We concatenate all postings in a thread to make a bag-of-word language model for the thread. We then retrieve the top N ranked threads by query likelihood scores computed based on the language models. Then, for all postings in the threads, we build a graph in the same manner as the posting-based algorithm. Figure 6.1(b)



(a) posting-based graph



(b) thread-based graph



(c) thread-based graph with thread structure

Figure 6.1. Graphs by different construction methods. A circle is a candidate node and a square is a posting node. A number in each square is the identification number of a thread to which the posting belongs.

shows an example. We can see that the thread-based graph uses a different set of postings from the posting-based graph.

Now we go one step further and consider thread structures to consider relationships among postings. A thread structure is a conversational structure established by reply relations in a thread, as introduced in Chapter 5. In most online communities, many-to-many communication is usual in a thread, and accordingly, readers can be confused with who talks to whom, particularly in long threads. With thread structures, this problem is resolved because reply relations distinguish each context from others. Further, we may identify what discussion each participant leads to. Therefore, we hypothesize that thread structures help identifying influential postings. An author of the influential posting may be assumed to be an expert.

With thread structures, we can make posting-to-posting links with them. However, there is an issue about these links. In the posting-to-candidate links, the direction of the links from postings to candidates looks natural because the authorities are the candidates rather than the postings and a document can be considered as a citation from a candidate’s knowledge. On the other hand, the direction of posting-to-posting links is somewhat vague. If a parent-child posting pair has a question-answering relation, then the authority is the child. On the other hand, if the pair has a suggestion-agreement relation, then the parent is likely to be authoritative. Even in a collection, there can be various relations. Therefore, we report results for parent-to-child as well as child-to-parent relationships in the experiments.

6.1.3 Expertise Ranking

For expertise ranking, we use a random walk algorithm similar to the PageRank algorithm [69, 95]. To customize the PageRank algorithm, we make a modification. A random walk matrix of the PageRank is defined as follows:

$$\bar{\bar{\mathbf{P}}} = \alpha \bar{\mathbf{P}} + (1 - \alpha) \mathbf{e}\mathbf{e}^T/n \tag{6.1}$$

where \mathbf{e} is the column vector of all ones, n is the order of the matrix, $\bar{\mathbf{P}}$ is an adjacency matrix where rows of dangling nodes are replaced by \mathbf{e}^T/n , and α is a parameter to control the effect of random jumps. The second term $\mathbf{e}\mathbf{e}^T/n$ is a random jump matrix in order to make the random walk matrix irreducible, which is a necessary condition for convergence of the PageRank vector.

We modify this random jump term. First, we prohibit random jumps between heterogeneous nodes, i.e., posting-to-candidate or candidate-to-posting. When considering a random surfer, jumps between documents sound natural. Further, jumps between candidates can be understood as communication outside the forum. However, posting-to-candidate can be considered as somewhat weird behaviors such as random authorship. We would like to avoid these jumps. Second, when reading a posting, a random surfer is likely to read other postings in the same thread because a user view usually displays multiple postings in a thread. That is, the probability of jump to postings in the same thread is possibly higher than that of jump to any other postings. Therefore, we consider a new random jump matrix as follows:

$$\mathbf{E}_{ij} = \begin{cases} 1/|V_C| & \text{if } i, j \in V_C \\ \beta/|V_{T_k}| + (1 - \beta)/|V_D| & \text{if } i, j \in V_{T_k}, \exists k \\ (1 - \beta)/|V_D| & \text{if } i, j \in V_D \\ 0 & \text{otherwise} \end{cases} \quad (6.2)$$

where V_C is a set of candidate nodes, V_T is a set of posting nodes in any thread, V_D is a set of posting nodes, and β is a parameter. This matrix is illustrated in Figure 6.2.

This new matrix is used for Equation (6.2) instead of $\mathbf{e}\mathbf{e}^T/n$. The final random walk matrix is stochastic and irreducible because nodes are fully reachable between candidates or postings, each posting is reachable from candidates by substitutions for dangling nodes, and a candidate has at least one incoming edge. Therefore, this

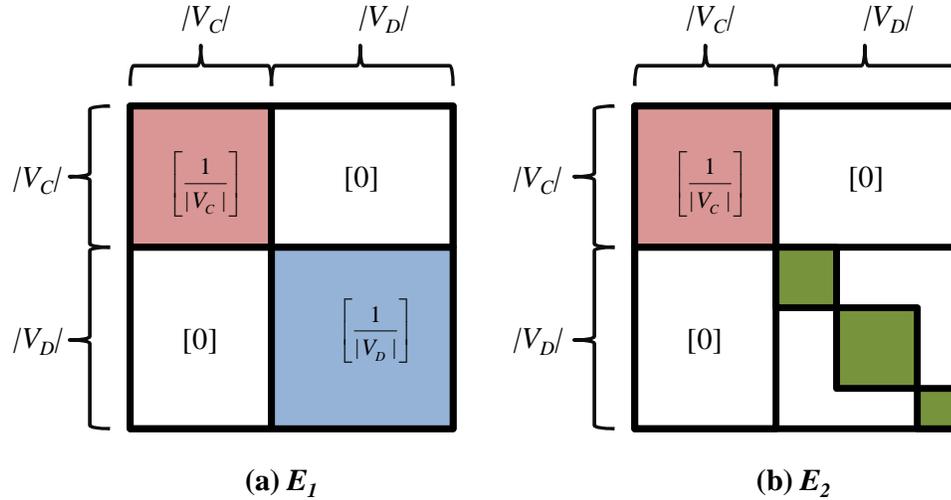


Figure 6.2. Two components of new random jump matrix for integrating hierarchical structures into the PageRank algorithm. These two matrices are linearly combined by β , i.e., $\mathbf{E} = (1 - \beta)\mathbf{E}_1 + \beta\mathbf{E}_2$. The red cells indicate random jumps among candidates while the blue cells indicate random jumps among documents (postings). The green cells indicates random jumps within a thread.

matrix guarantees a convergence of the PageRank vector. Both parameters α and β are set to 0.85 which is known as a magic number in the PageRank studies [69].

6.2 Experiments

We conduct experiments on two different types of collections: an email archive and a forum.

6.2.1 Email Archive

Email archives or newsgroups are old-style online communities but are still active in technical areas. We used the the ‘lists’ sub-collection of the W3C collection which has been used in Chapter 5.

Since the W3C collection has been used for the expert finding task of the TREC enterprise track 2005 and 2006 [124], there is a relevance judgment set provided by TREC. Since topics for TREC 2005 were used for the pilot evaluation and there is no manual judgment for them, we used only topics for TREC 2006, which contains

49 queries and 8,351 relevance judgments. Since thread structures provided in the W3C collection are inaccurate, we used the thread structures predicted by the thread structure discovery technique in Chapter 5.

To build a posting-based graph, we retrieved top 1,000 postings. The Dirichlet smoothing parameter was set to 450 that is the average length of a posting. Authorship information was extracted from ‘From’ field of each message. Using these postings and author information, a posting-based graph for each topic was constructed.

Note that we did not use the ‘To’ or ‘Cc’ fields to extract authors. Since the W3C collection was collected from an email archive, such fields exist. However, generally, most online communities provide only author information and postings are broadcast to all community members. To simulate this situation, we consider only authors in graph construction.

The same process was employed to build a thread-based graph. The differences are that the Dirichlet smoothing parameter was set to 1000 that is the average length of a thread and top 500 threads were retrieved for each topic because 500 threads include the similar number of authors as the 1,000 postings, i.e., approximately 2,000 authors. For thread structure-based graph, the reply relations inferred by thread structure recovery were used.

Results of expertise ranking are reported using two metrics: Mean Average Precision (MAP) and precision at top 5 (P@5). We considered a judged document whose relevance grade is equal to 2 as relevant. Table 6.1 presents the results.

All the thread-based methods show better performance than the posting-based method. Particularly, thread-structure based methods outperform the posting-based method. Further, the thread structure-based technique using the direction of child-to-parent is significantly better than the thread-based method. The change of performance depending on the direction of posting-to-posting edges is not noticeable.

Table 6.1. Expert finding results for different graph construction methods on the W3C collection. ‘Posting’, ‘Thread’ and ‘Thread Structure’ represent the posting-based, thread-based, and thread structure-based graph construction methods, respectively. (c→p) and (p→c) mean the direction of child-to-parent and parent-to-child for posting-to-posting edges. Superscripts α and β indicate statistically significant improvements on ‘Posting’ and ‘Thread’, respectively. (the paired randomization test with p -value < 0.1)

	MAP	P@5
Posting	0.2607	0.5306
Thread	0.2759 $^{\alpha}$	0.5429
Thread Structure (c→p)	0.2778 $^{\alpha\beta}$	0.5592 $^{\alpha\beta}$
Thread Structure (p→c)	0.2757 $^{\alpha}$	0.5592 $^{\alpha\beta}$

6.2.2 Forum

The second collection is an online forum collection. However, building test collections for expert finding is known to be very expensive even compared to building test collections for ad-hoc retrieval. This is because annotators should judge relevance by reading a number of documents written by an author or should be members of the community so that they can easily recognize the experts. To avoid this difficulty, we employed an automated test collection generation trick.

The Apple Discussions¹ provides separate forums for each product by Apple, Inc. Since these forums are divided by fine-grained categories, we can assume that each forum addresses a topic. That is, we consider an individual forum as a topically relevant thread set. We chose 30 forums so that the topics are as disjoint as possible. Table 6.2 shows examples of the chosen forums. From each forum, we crawled 30 randomly selected pages. Since each page contains 15 threads, we obtained 450 threads in total. Further, each forum of the Apple Discussions provides a top 10 user list based on points which are calculated by the number of replies and the quality of user feedback. We used this list as the gold standard for evaluation.

¹<http://discussions.apple.com/>

Table 6.2. Examples of the Apple Discussion forums used for the test collection

Product Category		Forum Title
iPhone	>	Phone
iPod shuffle	>	Using iPod shuffle (Second Generation)
iWork '09	>	Keynote '09
Safari	>	Safari for Mac

Forums in the Apple Discussions support the threaded-view, that is, each thread page displays reply relations among postings by indentations. Since this information is embedded in HTML tags, we can easily extract the reply relations by manually crafted rules.

Given that crawled forums are relevant thread sets, we can construct only thread-based graphs. Therefore, in this section, we do not compare thread-based methods to posting-based methods. Rather, we investigate effectiveness of different thread-based methods. Therefore, we constructed a thread-based graph for each topic (or forum), and we used the extracted reply relations for a thread structure-based graph.

Since we have only the top 10 users for each forum, it is not reasonable to treat all users behind top 10 as novices. Instead, we use recall-based metrics rather than precision-based metrics to observe how well the top 10 users are identified. We report recall scores at 10, 20 and 50 ($R@10$, $R@20$ and $R@50$). Table 6.3 shows the results.

The thread structure-based method using the direction of parent-to-child for posting-to-posting links outperforms the thread-based method. On the other hand, using the direction child-to-parent, the thread structure hurts performance. This suggests that the Apple forums are considerably biased toward the posting relations where replies usually have the authorities, e.g., question-answering relations. Therefore, depending on the characteristics of online communities, the choice of the direction of links between postings can be critical.

Table 6.3. Expert finding results for different graph construction methods on the Apple forums. Superscripts α and β indicate statistically significant improvements on ‘Thread’ and ‘Thread Structure (c→p)’, respectively. (the paired randomization test with p -value < 0.05)

	R@10	R@20	R50
Thread	0.6667 ^{β}	0.8367 ^{β}	0.9500 ^{β}
Thread Structure (c→p)	0.6500	0.8167	0.9300
Thread Structure (p→c)	0.6933 ^{$\alpha\beta$}	0.8600 ^{$\alpha\beta$}	0.9633 ^{$\alpha\beta$}

6.3 Conclusions

We addressed identification of an influential social class, i.e., experts in online communities. Specifically, we introduced how to define social structures. Based on these definitions, we proposed expertise graph construction methods and a variation of a random walk algorithm for expert finding. Using two different online community collections, we demonstrated that integration of social structures with other structures such as thread structures can be helpful for expert identification. In addition, we found that relations between graph nodes need to be differently considered depending on applications.

CHAPTER 7

SEARCH USING THREE STRUCTURES

We have introduced retrieval techniques using three core structures in social applications, i.e., hierarchical, conversational and social structures. More specifically, we have addressed representation techniques to construct retrieval objects containing relevant information along these structures. Indeed, although the representation techniques are similar to each other in that they are based on our geometric representation technique, each has been addressed individually. Since each structure encodes different aspects of social applications and a single structure cannot reflect various properties of the applications well enough, we may expect that retrieval performance could be improved if these structures can be represented or combined in a single framework. Therefore, in this chapter, we summarize our representation techniques exploiting each individual structure and introduce how to combine them. Also, this approach is evaluated on a forum search task where the three structures play crucial roles.

7.1 Representation Combining Three Structures

We begin with hierarchical structures. Indeed, other structures can be converted into hierarchical structures as we will see. A hierarchical structure is defined by ownership or containment relations. For example, the followings make hierarchical structures: a thread and its postings, and a blog site and its postings. In particular, when a retrieval object is a set object such as a thread or a blog site, we can have two different representations.

First, the object can have a coarse-grained representation by collapsing the boundary structure of its members, e.g., posting, and using the whole content. Assuming Dirichlet smoothing, its language model representation is given by

$$tf_{w,T} = \sum_{e \in T} tf_{w,e}$$

$$Pr(w|T) = \frac{tf_{w,T} + \mu \cdot cf_w / |C|}{\sum_w tf_{w,T} + \mu}$$

where T is a set object, e 's are its members, w is a word, μ is a Dirichlet smoothing parameter, and C is the collection. From this representation, we have the following ranking function:

$$\Gamma_{GR}(Q, T) = \prod_{q \in Q} Pr(w|T) \quad (7.1)$$

We call this the global representation.

Second, the object can have a fine-grained representation by preserving the boundary structure of its members and combining several selected members. For this representation, there can be many variations depending on the ways of selecting and combining the members. In this work, our focus is to locate relevant information. Accordingly, the members are selected according to their relevance estimates, e.g., their query likelihood scores. To combine individual representations of these members, we use the geometric mean, as suggested in Chapter 3.

$$Pr(w|T) = \left(\prod_{e \in S_{T,K}} Pr(w|e) \right)^{\frac{1}{K}}$$

where $S_{T,K}$ is a set of K selected members, e.g., the top K members according to their query likelihood scores. A ranking function by this representation is given by

$$\Gamma_{PCS}(Q, S_T^K) = \left(\prod_{e \in S_{T,K}} Pr(Q|e) \right)^{\frac{1}{K}} \quad (7.2)$$

Note that this computation benefited from the product-based combination because we can compute this ranking function directly from query likelihood scores $P(Q|e)$'s of the members, not from the combined language model representation. The derivation is provided in Equation (4.2). Also, when there are less than K members, we use the upper bound approach in Equation (4.3).

The selected members can be seen as topically related documents because they appear relevant to a topic query. Accordingly, the representation of each retrieval object looks like a centroid of topical clusters. Because of the similarity to clustering, we call this method pseudo-cluster selection.

Global representation and pseudo-cluster selection complement each other because the former provides a global context of the object while the latter provides local evidence mined from local contexts defined by a query. Accordingly, these two can be combined as follows:

$$\Gamma_{GR}(Q, T)^{1-\pi} \cdot \Gamma_{PCS}(Q, S_{T,K})^\pi \quad (7.3)$$

In the previous chapters, we have showed that combinations of these two representations by a weighted product consistently achieve better performance.

Although these representation methods are designed for hierarchical structures, other structures can benefit from these methods. For example, in thread search using conversational structures, a thread can be represented by various contexts extracted from conversational structures. In other words, a thread is a retrieval object while dialogue contexts introduced in Chapter 5 can be seen as members of the object. That is, we can apply the same representation to a hierarchical structure converted from a conversational structure. In other words, a thread can be represented by the

geometric mean of relevant dialogue contexts. In online communities, social structures can be defined by authorship as seen in Chapter 6. If an author is a retrieval object, then postings created by the author can be members of the object. That is, we can represent an author by her postings. Also, once an author representation that we here call an author model can be obtained, we can replace the original representations of the author’s postings by the author model and aggregate the new posting representations to represent a thread. In this manner, we can make various thread representations based on different structures.

Formally, a thread can be represented in four different ways as follows:

$$Pr(w|T) = \frac{tf_{w,T} + \mu \cdot cf_w / |C|}{\sum_w tf_{w,T} + \mu} \quad (7.4)$$

$$Pr(w|T) = \left(\prod_{p \in S_{T,K_p}^p} Pr(w|p) \right)^{\frac{1}{K_p}} \quad (7.5)$$

$$Pr(w|T) = \left(\prod_{d \in S_{T,K_d}^d} Pr(w|d) \right)^{\frac{1}{K_d}} \quad (7.6)$$

$$Pr(w|T) = \left(\prod_{p \in S_{T,K_p}^p} Pr(w|A(p)) \right)^{\frac{1}{K_p}} \quad (7.7)$$

where p and d are a posting and a dialogue context respectively, and S^p and S^d are sets of relevant postings and dialogues respectively. $A(p)$ is the author of posting p ; thus, $Pr(w|A(p))$ is an author model that is obtained by

$$Pr(w|A) = \left(\prod_{p \in S_{A,K_a}^p} Pr(w|p) \right)^{\frac{1}{K_a}}$$

where A is an author, and S_{A,K_a}^p is a set of K_a high query-likelihood postings created by A .

Therefore, Equation (7.4) and (7.5) correspond to an hierarchical structure, whereas Equation (7.6) and (7.7) correspond to a conversational structure and a social structure, respectively. Also, we can have a global representation ranking function from Equation (7.4). On the other hand, Equation (7.5), (7.6) and (7.7) are used for pseudo-cluster selection ranking functions.

Finally, we can combine these through an extension of Equation (7.3).

$$\Gamma_{GR}(Q, T)^\alpha \cdot \Gamma_{PCS}(Q, S_{T, K_p})^\beta \cdot \Gamma_{PCS}(Q, S_{T, K_d})^\gamma \cdot \Gamma_{APCS}(Q, S_{T, K_p})^{(1-\alpha-\beta-\gamma)} \quad (7.8)$$

where Γ_{APCS} is a pseudo-cluster selection ranking function based on the representation of Equation (7.7). Combination parameters, α , β and γ can be learned via various learning to rank techniques.

Figure 7.1 and 7.2 illustrate this representation. That is, we try to find a representation point of a thread by contexts extracted from each structure. In turn, we find a centroid of these representation points as the final representation of the thread.

7.2 Experiments

A forum search task, specifically, thread search, is a proper task for evaluating our approach combining three structures because all three structures clearly exist and play crucial roles for organizing information and encouraging people to participate in community activities. Accordingly, we use the same settings and collections that we used in Chapter 5. However, the collections do not have explicit conversational structures, i.e., thread structures, because they were obtained from flat-view forums. Although we predicted thread structures using our proposed algorithm, we cannot rule out the possibility that inaccurate thread structures affect our evaluation. Indeed, in Chapter 5, our goal is to provide a reasonable thread structure discovery algorithm and show how helpful thread structures are for retrieval performance. By

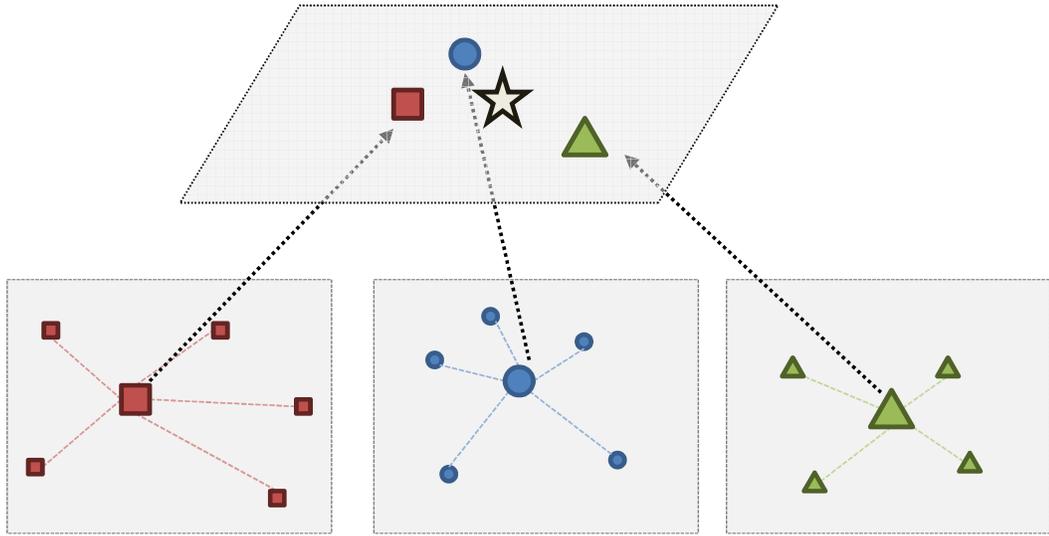


Figure 7.1. In the bottom planes, a small square, a circle and a triangle represent a posting, dialogue context and an author model, respectively. Large shapes denote their geometric mean representations. By Equation (7.8), we find a mean representation of these mean representations as shown in the upper plane.

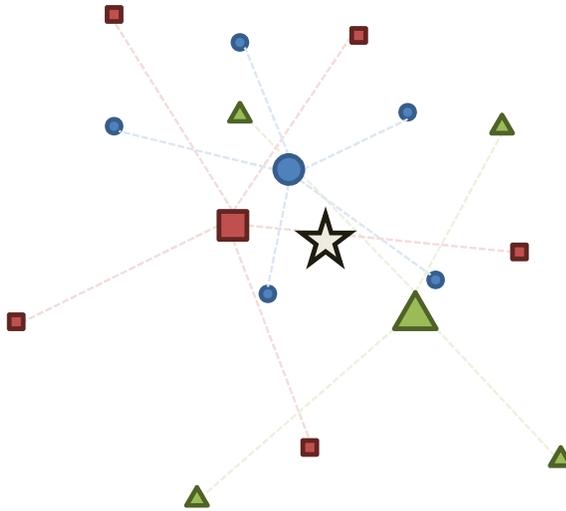


Figure 7.2. Illustration of 7.1 in a single plane.

comparing inaccurate thread structures and relatively accurate thread structures estimated by our algorithm, we could find clear evidence that accurate thread structures help retrieval. However, in this chapter, we do not focus on the accuracy of thread structure discovery. Therefore, we construct a new test collection which is free from any issues related to the accuracy of thread structures. In particular, this collection is built using crowd-sourcing. We first describe how to make this collection. Then, we report evaluation results on the previous two forum collections, i.e., WOW and CANCUN, as well as the new collection.

7.2.1 Constructing a test collection by crowd-sourcing

Making test collections in Information Retrieval is an expensive task. Collecting documents is relatively easy because a great volume of documents is readily available on the Web. If we have enough resources to crawl and store many documents, we can acquire a document set as large as we want. However, making relevance judgments still requires expensive human labor. As an alternative, many IR researchers have recently paid attention to crowd-sourcing [123, 49, 108, 2]. In particular, Amazon Mechanical Turk¹ is used for many annotation tasks including relevance judgments. This tool leverages the “cheap” labor of anonymous untrained people. Also, since many people can simultaneously participate in an annotation task, this task can be completed very quickly. That is, these annotation tools based on crowd-sourcing have great advantages in terms of cost and speed.

In this chapter, we used Amazon Mechanical Turk to obtain relevance judgments for a new test collection. First, we crawled 18 popular sub-forums of `Whiteblaze.net`² which is not only a forum for Appalachian trail thru-hikers but also one of the biggest hiking forums for general hiking information such as backpacking tips, trail informa-

¹<https://www.mturk.com/>

²<http://whiteblaze.net/forum/>

Table 7.1. Statistics of the Whiteblaze.net collection

#Threads	#Postings	#Postings per Thread	Avg. Posting Length (in words)
31,075	523,988	16.9	79.5

Table 7.2. Examples of queries for Whiteblaze.net

day hiking trails around Boston
LiteShoe’s The Ordinary Adventurer book review
poison ivy information
water source condition between Damascus to Erwin
tricks and techniques for ascents

tion, and gear reviews. In fact, since most people who make relevance judgments may not have experienced a forum that we select, we should be very careful when choosing a forum to be crawled. We chose the Whiteblaze.net forum because hiking is a very popular topic with which many people are usually familiar. On the other hand, if we had chosen a specific online game forum, most people, except the few who have played the game, could not have made relevance judgments. A summary of the Whiteblaze.net collection is presented in Table 7.1. This forum allows users to choose a post to which they reply. Since a thread structure is displayed in each thread page, we could extract the thread structures of all threads by performing rule-based parsing for thread pages.

To establish a test dataset, we randomly sampled a set of threads and decided if each thread is informative such that any user may want to find the thread by querying. For example, threads where members say hello to each other are classified as uninformative. We discarded uninformative threads until 50 informative threads are obtained. We asked human editors to make a query that they are likely to type to look for each informative thread. As a result, we obtained 50 queries. Table 7.2 shows several examples queries. Next, we made a relevance judgment pool using thread search techniques introduced in Chapter 5 so that a query has 50 threads to be judged. From this pool, we generated Hits which are unit tasks used in Amazon

Mechanical Turk. We designed Hits so that a simple guideline, a query and five threads to be judged are provided to Hit workers. Each judgment was performed on a three-point, i.e., non-relevant (0), relevant (1) and highly relevant (2). Also, we duplicated the threads in the pool so that each thread is judged by two different workers.

However, annotation results by crowd-sourcing are often very noisy because most participants are not only untrained but also unfamiliar with annotation tasks. Furthermore, in many cases, there are many spam annotations. Considering the anonymity of participants, this phenomenon is natural. Therefore, to address these drawbacks and acquire accurate annotations, some filtering techniques such as the trap method [108] have been proposed. We used the following rules to filter out workers who made noisy annotations.

1. Reject all Hits from a worker whose average time per judgment is shorter than 5 seconds.
2. Reject all Hits from a worker who made definitely wrong judgments. We know a highly relevant thread for each query because the query is generated from the thread. If the thread is judged as non-relevant, the judgment is almost certainly wrong.
3. Reject all Hits from a worker who made too many spurious judgments. We randomly sample several judgments by a worker and check their correctness in the following two cases: 1) Hits are too generous, e.g., highly relevant for most threads, and 2) too many Hits are left not being judged. These are signals that the worker is a spammer or a careless worker. If we decide randomly selected judgments are unreliable, we would reject all this worker's Hits.

Whenever all Hits were completely judged by workers, we applied the rule to the submitted Hits. In most cases, about 80% of Hits were determined as noisy

and rejected. We re-posted the rejected ones and repeated this step until we could obtain reliable judgments for all threads in our pool. Each cycle takes 3 or 4 days on average. We repeated this cycle 6 times. Although each cycle was done quickly, we had to wait for 3 weeks to obtain all judgments. Considering that speed is one of the biggest advantages of crowdsourcing, this is a somewhat disappointing result. In total, 119 workers contributed to this test collection, and the average time per Hit was 184 seconds. Finally, since each thread has two judgments, we averaged the judgments to generate the final judgments. Also, to reduce possible noise further, we manually marked all the threads used for query generation "definitely relevant" to which we assigned 3 as the relevance value. To investigate how accurate the labels are, 200 of these judgments were randomly sampled and manually reviewed. As a result, 84% of them appeared plausible. This shows that reasonably accurate labels can be acquired through strict filtering rules and iterative steps.

7.2.2 Results

The first experiment was done on the same forum test collections used in Chapter 5. We used the same settings including the same splits for 10-fold cross validation. For learning the linear combination parameters of Equation (7.8), we used Rank SVM [57]. Although we also employed other learning to rank techniques such as AdaRank [137] and LambdaRank [17], Rank SVM showed the best performance among them. A slack variable for SVM was set to 0.1, and a linear kernel was used. Table 7.3 presents the results. Our new three structure combination shows consistently better results compared to the best result from the earlier experiments for all metrics.

Next, we conducted the second experiment using the new forum collection - the Whiteblaze.net collection. For comparisons, we employed other techniques effective in the earlier experiments in Chapter 5. The Dirichlet smoothing parameters for each context were estimated by the unsupervised estimation method (Appendix B).

Table 7.3. Results by the three structure combination on WOW and CANCUN. “Dialogue + Thread” is the best result from Chapter 5. A † indicates a statistically significant improvement on “Dialogue + Thread” (randomization test with p -value < 0.05).

	WOW		CANCUN	
	NDCG@10	MAP	NDCG@10	MAP
Dialogue + Thread (PCS + GR)	0.4823	0.4073	0.5141	0.2973
Three Structure Combination	0.4855	0.4126	0.5351 [†]	0.3221 [†]

Table 7.4. Results by the three structure combination on Whiteblaze.net. A † indicates a statistically significant improvement on “Thread” (randomization test with p -value < 0.05).

	NDCG@5	NDCG@10
Thread (GR)	0.5656	0.5547
Dialogue (PCS)	0.5753	0.5704
Dialogue + Thread (GR + PCS)	0.5888	0.5761
Three Structure Combination	0.5903 [†]	0.5823 [†]

Also, we used RankSVM for learning the combination parameters. Leave-one-out cross validation was performed. We used NDCG at 5 and at 10 as evaluation metrics to reflect the multi-scale judgments. Table 7.4 presents the experimental results. The results show almost the same trend as appeared on WOW and CANCUN. The local context-based method (PCS) is better than the global context-based method (GR), and their combination (GR + PCS) outperforms each context-based method. Also, the three structure combination technique demonstrates the best performance. Although the performance differences are not so dramatic, the improvements are consistent.

These two sets of experiments show that we can benefit from appropriate combinations of social media structures. Also, retrieval techniques for social applications can be further enhanced by incorporating more social media structures.

7.3 Conclusions

To exploit social media structures further, we proposed a new representation and combination technique based on various techniques introduced in previous chapters. Also, to build a new test collection, we collected relevance judgments via crowdsourcing. Experiments on two previous forum collections and a new forum collection demonstrated that we can improve retrieval performance using three core structures simultaneously in our framework.

CHAPTER 8

IDENTIFYING RELEVANT SUBSTRUCTURES

All tasks that we have addressed so far involve retrieving objects that contain relevant information. A common assumption with these tasks is that users would be able to easily discover information relevant to their needs in the objects, once top-ranked objects are identified. For example, the retrieved objects of thread search are threads. For a short thread with several postings, users can easily find a piece of information that they need by reading through all postings. However, a thread can sometimes contain many postings, e.g., more than 50. Reading such a long thread to find relevant information is tedious even if we can assure that there is relevant information in the thread. Moreover, a thread often contains multiple conversations discussing different sub-topics. In this case, understanding such conversations may not be easy.

Therefore, in this chapter, we address the identification of relevant substructures in retrieval objects so that users can directly obtain relevant information without reading all contents of the objects. Specifically, we focus on relevant substructures in threads because we can exploit the structures in threads and relevant information tends to be contained over multiple postings rather than in a single posting, as we will see later.

We first discuss how to estimate posting-level relevance scores using language modeling approaches. Then, we introduce two techniques to select relevant substructures by maximizing substructure-level relevance incorporating posting-level relevance and

thread structures. Our algorithms are evaluated via experiments on a real forum collection. We also discuss construction of the test collection and an evaluation metric.

8.1 Related Work

Our task is similar to text snippet extraction or topic-based text segmentation in that we also assume a scenario that relevant fragments are extracted from a returned relevant document. For example, TextTiling by Hearst [50] is a well-known technique that considers shifts of subtopics in text representations. Also, Ponte and Croft [100] and Salton et al. [107] have done seminal work for text segmentation. In a broader sense, passage retrieval [18, 61] and XML retrieval [45] can be considered as a relevant line of research because relevant passages or elements can be relevant fragments. However, passage retrieval does not usually consider a specific document. On other hand, document summarization [48, 87] can be thought of as similar to our task because a few sentences are usually extracted from a specific document. However, many summarization studies do not focus on relevance to a query.

All these studies are different from our approach for the following reasons. Much of previous work focuses on how to obtain fragments. However, in this work, a thread consists of postings. That is, appropriate fragments, i.e., posting are given. More importantly, we incorporate contexts extracted from social media structures. That is, we seek an optimal substructure given thread structures and contexts.

8.2 Estimating Posting-level Relevance

Considering a thread as a set of postings, we want to identify a relevant subset of postings. This is formally defined as follows:

Input: query q , integer k and thread $T = \{p_1, \dots, p_n\}$

Output: $\arg \max_S R(q, S)$ s.t. $S \subseteq T$ and $|S| \leq k$.

where k is the number of postings that users are willing to read, and R is a relevance function mapping to a real value.

This formulation contains a set-level relevance function. However, how to define set-level relevance is somewhat unclear. Therefore, in what follows, we first estimate posting-level relevance, i.e., how relevant each posting p_i is to q . We then address finding the best subset using estimated posting-level relevance as evidence.

8.2.1 Posting Query-likelihood

We may use a function of q and p_i pair derived from a standard unigram language modeling framework, i.e., query-likelihood, as relevance evidence as follows:

$$R(q, p_i) = Pr(q|p_i) = \prod_{w \in q} Pr(w|p_i) \quad (8.1)$$

where $Pr(w|p_i)$ is a smoothed language model of p_i . In this work, we use the Dirichlet smoothing to estimate the model.

8.2.2 Multi-context Interpolation

A posting is sometimes too short to be a sufficient textual representation. Furthermore, when a discussion can be understood in conversational context, the textual representation of a posting can be extremely compact. For example, consider a posting in a thread that asks a question, e.g., “what are the best boots for summer season hiking?”. Assume that our query is the same as the question. Since this question can be read by all forum members, a person who posts a reply to the question posting

often assumes that all readers have the same context and gives a short answer while omitting some important keywords, e.g., “I like the Moab series (model) of Merrel (brand)”. Although this reply posting is highly relevant to the original question (or query), this posting cannot be determined as relevant by the query-likelihood model because there is no overlap between the textual representation and the query representation.

To mitigate this problem, we incorporate contextual evidence into posting-level relevance. A thread has implicitly or explicitly thread structures by reply relations. We can extract various contexts from the thread structures as done in Chapter 5. Specifically, in this chapter, two types of contexts, i.e., posting contexts and pair contexts, are considered. To estimate posting-level relevance, evidence based on different contexts are interpolated as follows:

$$R(q, p_i) = Pr(q|p_i) + \sum_{\{a|p_i \in a\}} Pr(q|a)$$

where a is a reply pair.

This technique make an effect on neighboring nodes of a node with a high posting query-likelihood score. In Figure 8.1, if node A and E have high posting query-likelihood scores, even when node B does not, pairs $A - B$ and $B - E$ are likely to have high pair query-likelihood scores. Accordingly, node B can have a high posting-level relevance estimate. That is, this approach can be seen as smoothing posting-level relevance along paths in a thread structure.

8.2.3 Enhancement via Query Expansion

One of the causes of sparseness of posting-level relevance is mismatch between textual representations of queries and postings. We can try to solve this problem by enriching query representations. A typical way of doing this is query expansion. We perform query expansion using the relevance model [71]. In particular, we employ a

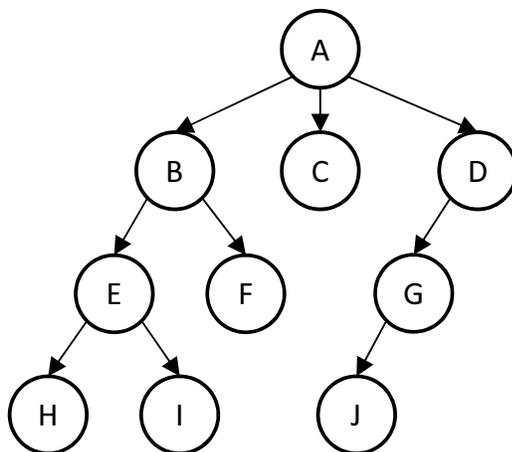


Figure 8.1. Example of a thread structure. An arrow represents a reply relation.

variant of the relevance model involving interpolation with the original query, which is often called RM3. This query expansion approach is not limited to any specific technique for estimating posting-level relevance; thus, we apply this approach to both techniques described earlier.

8.3 Relevance Maximization Through Thread Structures

Once we have estimated local relevance, i.e., posting-level relevance, we select a subset which maximizes global relevance, i.e., set-level relevance, using posting-level relevance as evidence. As explained in Chapter 8.2.2, simply selecting postings only with high posting-level relevance estimates is not sufficiently effective. For example, in threads consisting of question-answer postings, questions tend to be long while answers tend to be short. Since long postings probably have more query terms, we often select only question postings from threads in the worst case scenario. Therefore, we need to focus on more global contexts such as conversational structures embedded in thread structures rather than posting-level local evidence. Assuming that thread structures are able to give us a good guideline, we propose two techniques incorporating thread structures.

Input: k, L
Output: S
<pre> 1: $c \leftarrow k$ 2: $S \leftarrow \emptyset$ 3: for $i \leftarrow 0$ to $L - 2$ 4: $p_1 \leftarrow L[i]$ 5: for $j \leftarrow i + 1$ to $L - 1$ 6: $p_2 \leftarrow L[j]$ 7: if $route(p_1, p_2) \neq \emptyset$ and $route(p_1, p_2) \leq c$ 8: then 9: $S \leftarrow S \cup route(p_1, p_2)$ 10: $c \leftarrow k - S$ 11: fi 12: end 13: end 14: return S </pre>

Figure 8.2. Greedy Algorithm. S is a posting set to be return to users. k is the maximum size of S . L is a posting list sorted in descending order of posting-level relevance. $route(p_1, p_2)$ is a set of all postings on the route connecting p_1 and p_2 .

8.3.1 Greedy Approach

We may assume that consecutive utterances in a conversation consistently address similar topics. Under this assumption, if two highly relevant postings are connected through a route in a thread structure, all postings lying on the route would be relevant as well. Note that these two nodes should be connected in a directed acyclic graph (DAG) as shown in Figure 8.1. For example, node A and E are connected via a route $A - B - E$. However, node B and D are not connected as there is no route between B and E . In fact, these two nodes are in different branches each of which is assumed to make a separate conversation. Based on this assumption, we propose an algorithm as shown in Figure 8.2.

Although postings with the higher posting-level relevance estimates are considered earlier, early entry of highly relevant postings into S is not guaranteed due to other constraints. However, this algorithm operates like a greedy algorithm in that routes with highly relevant postings are included in S as long as the route meets another

constraint k . Even though a conversation contained in a long route ($> k$) may be highly relevant, the conversation would be incomplete if only k postings in the conversation are delivered to users. To avoid this situation, we reject all routes whose sizes are greater than k .

8.3.2 Mixed Integer Programming Approach

Another approach relaxes hard constraints by thread structures in contrast to the greedy algorithm. We convert such constraints into soft constraints as follows:

$$\begin{aligned} \text{Maximize:} \quad & \sum_i R(q, p_i) s_i - \sum_{i,j} D(p_i, p_j) s_i s_j \\ \text{Subject to:} \quad & \sum_i s_i \leq k \\ & s_i \in \{0, 1\} \quad \forall i \end{aligned}$$

where s_i is a binary variable indicating if p_i is included in S and $D(p_i, p_j)$ is the distance between p_i and p_j in a thread structure tree. The distance is computed as follows:

$$D(p_i, p_j) = 2 \cdot \text{depth}(LCA(p_i, p_j)) - \text{depth}(p_i) - \text{depth}(p_j)$$

where $\text{depth}(p_i)$ is the depth of p_i in the thread structure tree and $LCA(p_i, p_j)$ is the lowest common ancestor of p_i and p_j in the thread structure tree.

This formulation considers proximity as well as relevance. That is, it aims to find relevant postings located near each other in a thread structure. The constraint by thread structures is relatively weak in that direct connections among postings are not enforced. In fact, this problem setting is similar to sentence selection for document summarization [87, 46]. However, our work is different in that our quadratic term ($s_i s_j$) is related to distances measured in a thread structure instead of using the term for imposing a penalty on redundant sentences.

We can remove the quadratic term by introducing new variables s_{ij} 's so that the problem can be solved via mixed integer programming.

$$\begin{aligned}
\text{Maximize:} \quad & \sum_i R(q, p_i) s_i - \sum_{i,j} D(p_i, p_j) s_{ij} \\
\text{Subject to:} \quad & \sum_i s_i \leq k \\
& s_i \geq s_{ij} \quad s_j \geq s_{ij} \quad \forall i, j \\
& s_i + s_j - s_{ij} \leq 1 \quad \forall i, j \\
& s_i \in \{0, 1\} \quad s_{ij} \in \{0, 1\} \quad \forall i, j
\end{aligned}$$

We solve this problem using the simplex algorithm [96]. Note that both $R(\cdot)$ and $D(\cdot)$ are normalized.

8.4 Experiments

8.4.1 Forum Data

For evaluating the proposed methods, we used the Whiteblaze.net collection and the same topic set used in Chapter 7. As described earlier, each topic was made from a real thread. Therefore, we know a thread which contains relevant information to each query. We asked editors to rate each posting in the threads according to the degree of importance as part of a response to the corresponding query of the thread. As a result, each posting has a rating from 0 (Never include) to 3 (Must include). Note that the average number of postings for all threads in the collection is 16.9. However, the average number of postings for the threads used for evaluation is 19.6. This is because when the threads were selected, we filtered out uninformative threads which are usually short, as explained in Chapter 7.

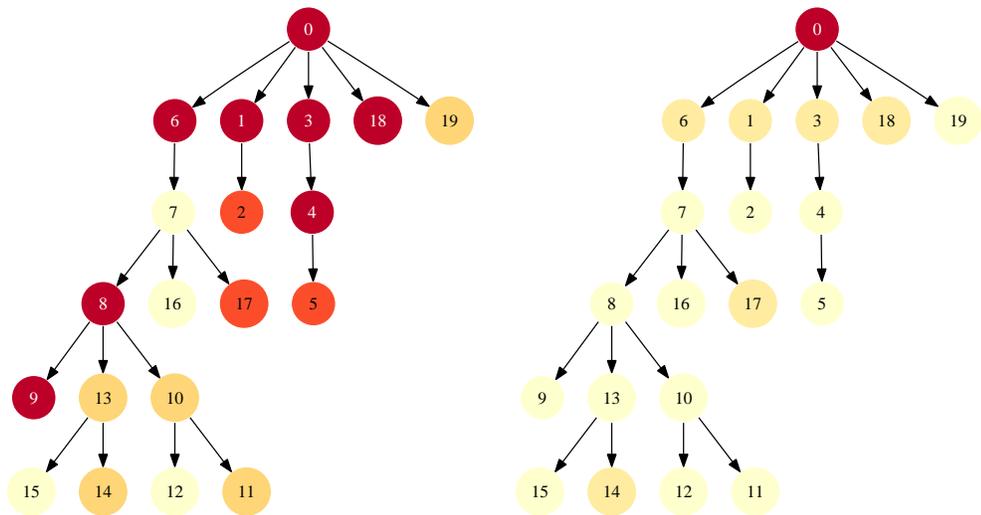
8.4.2 Evaluation and Baselines

When evaluating results by the proposed techniques as well as baselines, we need to be careful because of some unique properties of our task. First, our task has cutoff k . We assume that users are willing to read at least k postings. Therefore, we are interested in a set of k selected postings but not relative rankings of postings in the set. Second, threads have the different numbers of postings. Each thread should be treated differently according to its length. If the number of postings in a thread is less than or equal to k , we should not count the thread. Moreover, the benefit from recognizing k relevant postings in a very long thread would be greater than that in a thread which is marginally longer than k . Based on these aspects, we introduce a new evaluation metric as follows:

$$G_k = \frac{1}{Z} \sum_T I(l_T > k) (1 + \rho)^{(l_T - k - 1)} \sum_{p \in S_T} r(p)$$

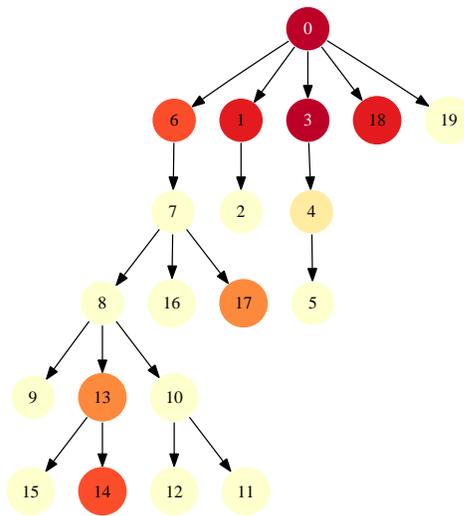
where T is a thread, l_T is the length (#postings) of T , S_T is a subset of T to be evaluated and $r(p)$ is a rating of posting p . ρ is a length bias parameter and set to 0.05 through our experiments. Z is a normalization factor and computed assuming the optimal subset which contains k postings with the highest ratings. We call this metric Normalized Length-Biased Gain (NLBG).

For comparisons, we employed two baselines. The first baseline assumes the scenario that only the link of a relevant thread is delivered to a user. If a user follows the link, the user would see postings in the first page of the thread which are sorted in chronological order of posting times. Therefore, the first baseline is selecting the first k postings in chronological order. The second baseline is selecting k postings according to their posting-level relevance, i.e., query-likelihood scores. This can be seen as a result of a posting search.



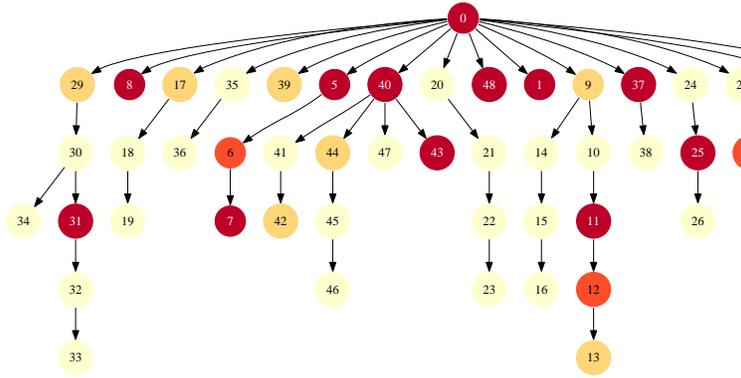
(a) Manually annotated relevance

(b) Estimated posting-level relevance by posting query-likelihood

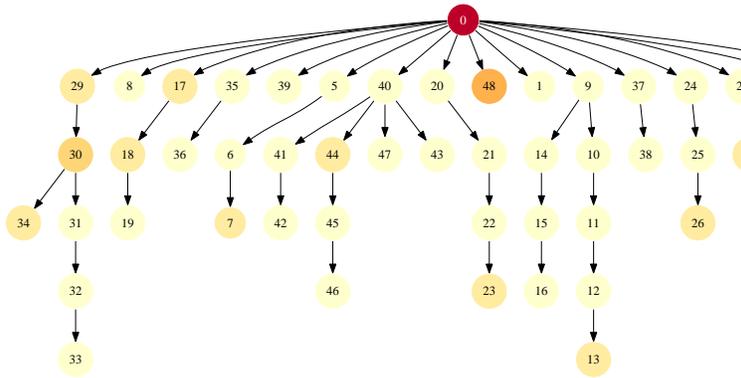


(c) Estimated posting-level relevance by multi-contexts interpolation

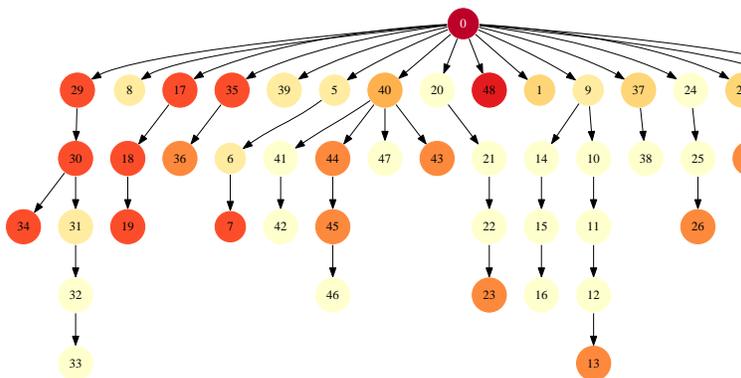
Figure 8.3. Relevant substructures [Thread ID = 7333]. The redder a node, the more relevant it is. A number in each node is the posting’s chronological order.



(a) Manually annotated relevance



(b) Estimated posting-level relevance by posting query-likelihood



(c) Estimated posting-level relevance by multi-contexts interpolation

Figure 8.4. Relevant substructures [Thread ID = 44226]

8.4.3 Results

Figure 8.3 and 8.4 present two examples of thread structures where postings are painted according to their real relevance or estimated relevance. We normalized manual annotation ratings or estimated relevance scores and mapped each normalized real value into a color. As you see, when using only posting contexts, the distribution of relevance is very sparse. On the other hand, when using the multi-contexts interpolation approach, the distribution is much smoother. Also, we can observe that the distribution becomes more similar to the ground truth by annotations.

Table 8.1 shows evaluation results of the various techniques, varying the cutoff from 2 to 6. The top two rows correspond to the two baselines. The third and fourth rows perform posting-level relevance estimation only by query-likelihood scores while the bottom two rows use multi-contexts interpolation as well. Our proposed techniques, “Greedy” and “MIP” outperform the baselines for all cutoffs. In particular, “MIP” consistently demonstrates better performance than “Greedy” except for cutoff 2. However, as the cutoff increases, the performance differences become small. Also, multi-contexts interpolation helps in all cases.

We also repeated the same experiments using posting-level relevance estimation enhanced by query expansion. We made thread documents of all threads in the collection by concatenating postings in each thread and built an index using Indri¹. Then, we retrieved the top 10 threads for each query using the unigram language model. Query expansion was performed by RM3 and the number of expanded query terms was set to 10. Using these expanded queries, we computed the query-likelihood scores of posting contexts and pair contexts. Table 8.2 presents the results. As we see, the performance is improved in almost all cases compared to the results without leveraging query expansion. Our proposed techniques still demonstrate the

¹<http://www.lemurproject.org/indri/>

best performance when combined with the multi-contexts interpolation approach. One difference is that “Greedy” shows better performance than “MIP” for more cutoffs. However, the difference is marginal, and they both achieve good performance.

8.5 Conclusions

We addressed the identification of relevance substructures in retrieval objects, specifically, forum threads. The proposed techniques incorporating posting-level relevance and thread structures demonstrated that they can identify relevant substructures accurately and also outperform simple result presentations by whole threads or individual postings. This shows that we need to focus on relevant substructures more to find optimal ways of delivering information to users.

Table 8.1. Evaluation results of proposed techniques and baselines (Chronological order and Posting-level Relevance order) according to different cutoffs when query expansion is not employed. “MIP” denotes the mixed integer programming approach. A bold number indicates the best performance for each cutoff. Since the number of the corresponding topics depending on the cutoff varies, the number are also reported.

Cutoff (k)	2	3	4	5	6
# of topics	50	49	49	47	44
Chronological order	0.7693	0.6691	0.6870	0.6776	0.6462
Posting-level Relevance order	0.7726	0.6827	0.7167	0.7014	0.6517
Greedy	0.8692	0.6681	0.6865	0.6381	0.6561
MIP	0.7744	0.7199	0.7214	0.7025	0.6817
Greedy with interpolation	0.9193	0.7744	0.7213	0.6958	0.7126
MIP with interpolation	0.8121	0.7950	0.7355	0.7238	0.7338

Table 8.2. Evaluation results of proposed techniques and baselines according to different cutoffs when query expansion is employed. A bold number indicates the best performance for each cutoff.

Cutoff (k)	2	3	4	5	6
# of topics	50	49	49	47	44
Chronological order	0.7693	0.6691	0.6870	0.6776	0.6462
Posting-level Relevance order	0.8128	0.7392	0.7537	0.7328	0.7102
Greedy	0.8949	0.7123	0.7337	0.6770	0.6882
MIP	0.8133	0.7467	0.7596	0.7372	0.7057
Greedy with interpolation	0.9542	0.8200	0.7986	0.7413	0.7173
MIP with interpolation	0.9162	0.8016	0.7900	0.7538	0.7319

CHAPTER 9

TEXT REUSE STRUCTURES AND TEXT REUSE PATTERN ANALYSIS

We have discussed retrieval techniques using three core structures in social media applications. However, there are other important structures which can help search in social applications. For example, text reuse structures imply strong relationships among text across many different social applications as well as many documents. Text reuse occurs when people borrow or plagiarize sentences, facts, or passages from various sources. For example, near-duplicate detection is one of the major applications that have been studied by numerous researchers since it can be used to achieve efficient search engines by getting rid of near-duplicate documents. Another obvious application involving text reuse is plagiarism detection. However, being able to detect local reuse would be a powerful new tool for other possible applications involving text analysis. For example, Metzler et al. [88] discussed tracking information flow, which is the history of statements and “facts” that are found in a text database such as news. This application was motivated by intelligence analysis, but could potentially be used by anyone who is interested in verifying the sources and “provenance” of information that they are reading.

Text reuse structures may not be directly related to search tasks if we consider only retrieval performance in a narrow sense. However, by investigating text reuse patterns, we can get insights for understanding user behaviors in social applications and leading to better application designs. Also, understanding these characteristics of social applications is essential in order to develop effective algorithms to maximally

leverage the potential that social applications have as information sources. For example, we can detect that a paragraph written by a specific user is frequently reused by other users in an online community. This may mean that the user is popular or authoritative in the community. That is, text reuse detection results can be leveraged for finding experts or designing enhanced ranking functions. Therefore, text reuse structures have the potential to be beneficial for search in social media applications.

In this chapter, we introduce algorithms for text reuse detection and analyze text reuse patterns on two social applications, i.e., blogs and microblogs.

9.1 Related Work

There have been broadly two approaches to text reuse detection. One approach is using document fingerprints through hashing subsequences of words in documents. This approach is known to work well for copy detection. Shivakumar and Garcia-Molina [117, 118] and Broder [14] introduced efficient frameworks. Since handling many fingerprints is too expensive, various selection algorithms for fingerprints were proposed by Manber [85], Heintze [51], Brin et al. [13] and Schleimer [109]. Broder et al. [15] suggested an efficient near-duplicate algorithm generating new fingerprints (super-shingles) by hashing sequences of fingerprints again. Charikar [22] introduced a hashing algorithm based on random projections of words in documents. Henzinger [52] empirically compared a variant of Broder et al's algorithm and Charikar's algorithm on a large scale Web collection. Chowdhury et al. [24] and Bernstein and Zobel [9] proposed filtration algorithms for fast near duplicate detection.

Another approach is computing similarities between documents in the Information Retrieval sense. Allan [1] addressed creating links connecting similar documents in his thesis. Shivakumar and Garcia-Molina [117] and Hoad and Zobel [53] suggested similarity measures based on relative frequency of words between documents. Met-

zler et al. [88] compared similarity measures using an evaluation corpus that was developed for studies of local text reuse.

There has been little work about near-duplicates or text reuse in social application domains, although Petrovic et al. [99] used near-duplicate detection algorithms for the first story detection task for Twitter.

9.2 Text Reuse Basics

We first provide a more detailed overview of text reuse to help readers understand the following text reuse pattern analyses.

9.2.1 Definitions of Text Reuse

Most text reuse detection algorithms are based on fingerprinting techniques in order to efficiently handle documents. For example, a document is segmented into multiple subsequence of words. In turn, each subsequence is converted into a fingerprint by hashing algorithms such as MD5 [105] or Rabin fingerprinting [102].

A text reuse relationship is a pairwise relationship. Given a pair of documents, we need to estimate the amount of text shared between the two documents. The amount of text of document A that is shared with document B can be represented as a ratio of the number of shared fingerprints to the number of fingerprints of document A. The ratio, *containment* of A in B [14] is estimated as follows:

$$C(A, B) = \frac{|F_A \cap F_B|}{|F_A|} \quad (9.1)$$

where F_A and F_B are sets of fingerprints of document A and B, respectively.

Note that the shared fingerprint ratio is a non-symmetric metric, i.e., $C(A, B) \neq C(B, A)$. Generally, symmetric metrics like *resemblance* [14] have been used for near-duplicate detection because it has to be determined whether the estimated value is

greater than a threshold in order to easily check if the document pair has a near-duplicate relationship. Since our goal is to understand more general forms of text reuse rather than simply judging near-duplicate documents, we use the non-symmetric metric that contains more information.

We divide containment values into three ranges as shown in Table 9.1. That is, if greater than 80%, 50% or 10% of the total fingerprints of document A are shared with a document B, then we say that *most*, *considerable* or *partial* text of document A is reused by document B. These thresholds are not fixed but may be changed based on the properties of collections or goals of the text reuse application. Here, we set the values based on reviewing results for various collections.

Table 9.1. Definitions of text containment terms

Term	<i>Most</i>	<i>Considerable</i>	<i>Partial</i>
Range	$C(A, B) \geq 0.8$	$C(A, B) \geq 0.5$	$C(A, B) \geq 0.1$

General text reuse occurs in various levels. Most of the text of a document might be shared with other documents, or only several words of a document might be shared with other documents. As a basis for evaluating the frequency of text reuse, we classify text reuse relationships into six categories as shown in Table 9.2. For example, if *partial* text of document A is shared with document B and the shared text is *most* text of document B, then document A and document B have a *C3* type relationship.

Table 9.2. Text Reuse Categories

Term	Relationship
<i>C1</i>	<i>Most-Most</i>
<i>C2</i>	<i>Most-Considerable</i>
<i>C3</i>	<i>Most-Partial</i>
<i>C4</i>	<i>Considerable-Considerable</i>
<i>C5</i>	<i>Considerable-Partial</i>
<i>C6</i>	<i>Partial-Partial</i>

Note that in a broad sense, $C1$, $C2$ and $C4$ correspond to near-duplicate cases, whereas $C3$, $C5$ and $C6$ correspond to local text reuse. We now briefly describe each category or type.

- $C1$ (*Most-Most*): This is a typical near-duplicate case, where two documents are almost identical.
- $C2$ (*Most-Considerable*): Generally, in this case, a short passage is added to text of another document. A typical example can be observed in blogs, i.e., copying the entire text of a news article and appending a short comment about the article.
- $C3$ (*Most-Partial*): In this case, a whole document is used as a partial text of a new document. $C3$ types are typically shown in cases where a news article is composed of several small news articles or where a document quotes interviews from other short news articles.
- $C4$ (*Considerable-Considerable*): This is a case where a new document is composed of large parts of other documents.
- $C5$ (*Considerable-Partial*): This is generally similar to $C4$ except for the amount of the shared text.
- $C6$ (*Partial-Partial*): This generally happens with boilerplate text or common phrases.

9.2.2 Text Reuse Detection

For efficient text reuse detection, an inverted index is generally built with fingerprints extracted from documents. To find all documents which have text reuse relationships with a document A , we first read all inverted lists of the fingerprints of document A , then merge the lists, and finally, find text reuse relationships. The

first step is the most critical in time complexity because it requires significant I/O access, whereas the other steps can be performed in main memory. Since the maximum length of the inverted list is the number of documents in the collection, this can be naively thought as an $O(Mn)$ algorithm, where M and n are the number of the fingerprints of document A and the number of documents in the collection, respectively.

On real collections, however, the length of the inverted list is at most the occurrence count of the most frequent fingerprints in the collection. Moreover, we can restrict the upper bound of the length by setting very common fingerprints to stop-fingerprints in the same way as stop-words in Information Retrieval. Therefore, the practical time complexity is $O(Ml)$, where l is the restricted length of the inverted list such that $l \ll n$.

When we try to discover all text reuse relationships in the collection, the above process is repeated n times, where n is the number of documents in the collection. This is an $O(nml)$ algorithm, where m is the average number of the fingerprints of a document.

9.3 Text Reuse Pattern Analysis in Blogs

We present a robust fingerprinting technique for text reuse detection, i.e., “DCT fingerprinting”, and analyze text reuse in blogs using the method.

9.3.1 DCT fingerprinting

We first split text into a few meaningful text segments such as phrases or sentences. In particular, we use the hash-breaking technique [13] which computes hash value $h(w)$ for each word w and selects hash values such that $h(w) \bmod p \equiv 0$ as breakpoints for text segments. That is, a sequence of words from the next word of the previous breakpoint to the current breakpoint is considered as a meaningful text segment.

We can apply a robust method called *DCT fingerprinting* to these text segments. The Discrete Cosine Transform (DCT) is a real valued version of Fast Fourier Transform (FFT) and transforms time domain signals into coefficients of frequency component. By exploiting a characteristic that high frequency components are generally less important than low frequency components, DCT is widely used for data compression like JPEG or MPEG. DCT is formulated as follows:

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad (9.2)$$

$$k = 0, 1, \dots, N - 1$$

where x_n and X_k are the n^{th} value in the time domain signal sequence and a coefficient of the k^{th} frequency component, respectively. Note that the length of the time domain signal sequence N is the same as the number of the frequency domain components.

A main idea of DCT fingerprinting is that a sequence of hash values of words can be considered as a discrete time domain signal sequence. That is, we can transform the hash value sequence into the coefficients of frequency components by using DCT.

The process of DCT fingerprinting is composed of seven steps as shown in Figure 9.1.

DCT fingerprinting is expected to be more robust against small changes than hash-breaking. As shown in Equation (9.2), when there is a small change of an input value, i.e., a hash value of a word, the change is propagated over all coefficients by a reduced effect. Since we quantize the coefficients, the final fingerprint value can be kept unchanged. That is, this robustness can be interpreted as an advantage of data reduction. Examples in Table 9.3 show the robustness of DCT fingerprinting. The numbers in [] are the fingerprints for the right string sequences.

It is difficult to show theoretically how many changes DCT fingerprinting can be tolerant of because input signal values are almost randomly mapped to by hashing.

1. Get a text segment by using revised hash-breaking with a parameter p .
2. Compute hash values for words in the text segment, x_0, x_1, \dots, x_{N-1} , where N is a length of the text segment.
3. Perform a vertical translation of the hash values so that the median of the hash values is located at 0.
4. Normalize the hash values by the maximum value.
5. Perform DCT with the normalized hash values.
6. Quantize each coefficient to be fit in a small number of bits, e.g., 2, 3 or 4 bits.
7. Form a fingerprint with the quantized coefficients Q_k 's as shown in Figure 9.2. If N is so big that all Q_k 's cannot fit the format, use only lower frequency coefficients. One approach is to use only the p lower frequency coefficients if the length of the text segment N is greater than the hash-breaking parameter p .

Figure 9.1. DCT fingerprinting

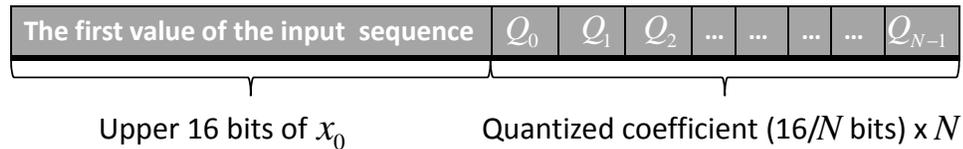


Figure 9.2. A format of 32bit DCT fingerprint

That is, while a minor change, e.g., a change from ‘product’ to ‘products’ might cause a big change of the hash value of the word, a word replacement might be coincidentally mapped to the same value. Nevertheless, a single word change tends to change a few high frequency components, and we can ignore the high frequency components by the formatting scheme. Thus, we can expect that DCT fingerprinting sometimes handles a single word change. When more than one word is changed, the input signal shape is likely to be distorted and the DCT coefficients are changed. Moreover, if words are added to or removed from the text segment, then even the number of the coefficients is changed. Therefore, we conclude that DCT fingerprinting can be tolerant of at most a single word replacement.

Comparisons among DCT fingerprinting and other popular methods including 0 mod p [85], winnowing [109] and hash-breaking [13] have been presented in [111],

Table 9.3. Examples of robustness of DCT fingerprinting

[0x295D0A52]	one woman comedy by person Willy
[0x295D0A52]	one woman show by person Willy
[0xF1315F87]	company scheduled another money
[0xF1315F87]	company slated another money

where DCT fingerprinting has been proved effective as well as efficient for text reuse detection.

9.3.2 Results and Discussions

We analyze the amount and type of text reuse on the TREC Blogs06 collection [83] which contains about 3 million postings, using DCT fingerprinting. We use two metrics to analyze the collection. One metric, the number of documents in each text reuse type, shows how many documents involve text reuse. Another metric is the average number of *siblings*. The siblings of a document represent documents which have text reuse relationships with the document.

When we find text reuse in blog collections, there is a problem to be considered. In most blogs, navigation bars are located on the top or the side of each page and advertisement links like Google AdSense¹ or links to the previous postings occupy the corners of each page. Text in such frames is repeated in most of the postings of a blog. As a result, blog postings could be falsely detected as text reuse relationships even though their actual contents are not related to each other at all. We refer to this as *frame noise*. To remove such noise, we employed a Document Slope Curve (DSC) content selection algorithm [38]. The algorithm plots a curve as follows:

$$DSC[k] = \begin{cases} 0 & \text{if } k = 0 \\ DSC[k - 1] + 1 & \text{else if } T[k] \text{ is a tag} \\ DSC[k - 1] & \text{otherwise} \end{cases} \quad (9.3)$$

¹<http://www.google.com/adsense>

Table 9.4. Text reuse detection results in TREC Blogs06 collection. ‘#Sibling’ represents the average number of documents which are related to the detected document through a category.

Type	#Doc	%	#Sibling
<i>C1</i>	125241	3.90%	562.16
<i>C2</i>	171619	5.34%	612.54
<i>C3</i>	166527	5.18%	731.68
<i>C4</i>	269064	8.38%	528.34
<i>C5</i>	439655	13.69%	688.60
<i>C6</i>	450539	14.03%	973.53
Total	675015	21.02%	1749.43

Table 9.5. Text reuse patterns in the TREC Blogs06 collection.

Pattern	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	Total
<i>Text Reuse</i>	16%	20%	20%	6%	12%	18%	15%
<i>Common Phrase</i>	2%	12%	12%	24%	28%	28%	18%
<i>Spam</i>	30%	22%	20%	8%	12%	20%	19%
<i>Frame</i>	36%	46%	48%	62%	48%	34%	46%
<i>URL Aliasing</i>	16%	0%	0%	0%	0%	0%	3%

where $T[k]$ is the k^{th} token in an HTML page. By exploiting the observation that there are fewer HTML tags in content bodies than in the other areas, we regard the lowest slope area of the curve as the content body.

Table 9.4 shows the text reuse detection results by DCT fingerprinting. Many more documents are involved in text reuse relationships. In fact, the numbers were overestimated as we see later.

We sampled 50 document pairs for each type from the detection results and manually classified the text reuse into three more classes based on the style of text reuse (rather than the amount of text). These classes are ‘*Text Reuse*’, ‘*Common Phrase*’, ‘*Spam*’, ‘*Frame*’ and ‘*URL Aliasing*’. The result is shown in Table 9.5.

‘*Text reuse*’ patterns correspond to actual text reuse cases. That is, a document pair with these patterns is derived from the same source or has a direct relation. Most text reuse originated from authoritative sources such as news articles or academic papers. This appears more frequently than text reuse based on other blog

postings. That is, many bloggers seem to still trust authoritative sources more than blog documents.

‘*Common phrase*’ patterns are caused by common phrases. Thus, we might not infer any actual relation. Most of these patterns are composed of boilerplate text. For example, the following paragraph is a representative example of boilerplate text which is located below content text with the highlighted date changed.

This entry was posted on **Friday, January 13th, 2006** at **12:00 pm** and is filed under XXX. You can follow any responses to this entry through the RSS 2.0 feed.

The boiler plate text forms a small part of the document, e.g., a header or footer rather than the content of the document. Also, this pattern is observed in most postings.

‘*Frame*’ patterns correspond to frame noise. Although we preprocessed the collection by using the DSC content selection algorithm, a considerable amount of frame noise still remains. Since this noise is almost evenly distributed over all types, we cannot distinguish it easily by classification.

Another new pattern is ‘*Spam*’. Spam phrases such as ‘free gift’ and ‘poker casino’ tend to be repeated in or between spam postings, and accordingly, they could be detected as text reuse.

Another special pattern is ‘*URL Aliasing*’ which has been reported in near-duplicate studies on Web [118]. While two postings have different URLs, they correspond to the same document. Since their contents are identical, these patterns are observed in only the C1 type.

As you see in Table 9.5, noisy patterns like ‘*Frame*’ and ‘*Spam*’ account for 50~70% of each class, which causes most of the overestimation of text reuse. There-

fore, to more accurately investigate text reuse in blog or Web collections, better content selection techniques and spam filtering algorithms are required.

In addition, ‘*Text Reuse*’ patterns are almost equally distributed over all text reuse types. That is, we need to consider all text reuse types in order to accurately infer relationships between documents. Therefore, for text reuse detection applications like information flow tracking, local text reuse detection is likely to be more effective than near-duplicate detection which can detect only a few text reuse types.

9.4 Text Pattern Analysis in Microblogs

We review a near-duplicate detection technique, and analyze text reuse in microblogs using the method.

9.4.1 Locality Sensitive Hashing (LSH)

Since microblogs allow only short text, we do not need to leverage techniques for general text reuse detection such as DCT fingerprinting. For example, in Twitter, a tweet should be fewer than 140 characters. Under this circumstance, differentiating text reuse categories may be meaningless. Rather, we use a near-duplicate detection technique which is much more efficient than general text reuse techniques because it takes only “Most-Most” (C1) text reuse into account.

For near-duplication detection, we use locality sensitive hashing (LSH) [22] which is one of the most widely used techniques for near neighbor search in high dimensional spaces. More specifically, we follow the practice introduced in [52].

Each word in document D is randomly projected into b -dimensional space where each coordinate is $[-1, 1]$. After adding all projected vectors corresponding to words in D , we set the i -th entry of the final vector to 1 if the i -th entry of the added vector is greater than 0, and to 0 otherwise. Accordingly, we have b -dimensional final vector representing D . The new representation is denoted by $\text{1sh}(D)$.

Once we have new vector representations for all documents in a collection, we find all document pairs between which the hamming distance is less than or equal to d . These pairs are called near-duplicate pairs. Since microblogs are much shorter than general web documents, we use $b = 64$ and $d = 1$ that are smaller compare to the values used in [52]. When $d = 1$, it is trivial to find near-duplicate pairs. That is, for given $\text{1sh}(D)$, we perform 1-bit perturbation and look up other documents matching the perturbed vectors. Finally, each found document and D make a near-duplicate pair.

9.4.2 Pattern Analysis

For analysis, we used a real microblog dataset which was collected from Twitter by Petrović et al. [99] for six months (April 2009 to October 2009). This collection contains 164 million time-stamped tweets. From the tweets, we made two subsets containing tweets only within two time spans, i.e., one day and one week, to see a difference between near-duplicate patterns according to the lengths of time spans. The start time of each time span was randomly selected. **DAY-SET** and **WEEK-SET** denote the subsets selected by the time spans, respectively. Since re-tweets are likely to be falsely considered as near-duplicates, we removed all text following “RT” from each tweet. In addition, tags for mentioning and labeling, i.e., tokens starting with # or @ were also removed.

Table 9.6 presents near-duplicate detection results for the two subsets. The results show that the portion of near-duplicates in **WEEK-SET** is larger than in **DAY-SET**. This may be because the time spans of some near-duplicates are actually over a day or hash collision occurs more frequently as the number of samples increases. The average time gap between two near-duplicate tweets is shorter than 2 hours. Indeed, as shown in Figure 9.3 the time gap distribution follows Zip’s law, and most time gaps are very short.

Table 9.6. Near-duplicate detection results for two time spans. “ND-detected” denotes tweets involved in at least a near-duplicate relation. “# ND-detected by the same users” means the number of tweets posted by the same users among “ND-detected”. “# ND per ND-detected” denotes the average number of its near-duplicate tweets per ND-detected tweet. “Time gap per ND-pair” denotes the average time difference between posting times of tweets which make a near-duplicate pair.

	DAY-SET	WEEK-SET
# Tweets	1.3M	9.5M
# ND-detected	92K	876K
# ND-detected by the same users	6.2K	92K
# ND per ND-detected	3.8	4.4
Time gap per ND pair	97min	120min

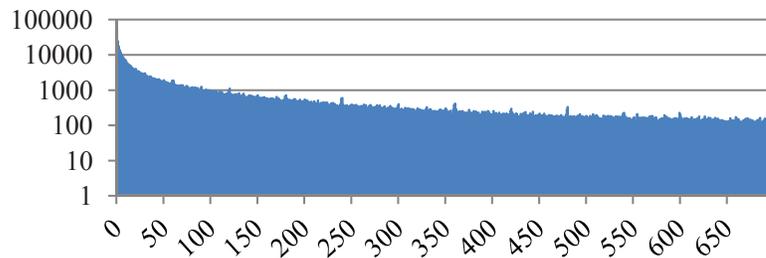


Figure 9.3. Distribution of time gaps of near-duplicate pairs in WEEK-SET. (x -axis: time gap (in minutes), y -axis: frequency)

For further analysis of near-duplicate patterns in microblogs, we manually classified 100 near-duplicate pairs randomly sampled from the results for WEEK-SET. We considered four categories: SPAM, COMMON, ISSUE and UNCLASSIFIED.

SPAM contains spam tweets such as advertisement campaigns and automatically generated text by software rather than by humans. Tweets that are consistently popular in Twitter or social networking tools are considered COMMON, e.g., chat abbreviations or greetings. ISSUE is the most interesting type. When an issue or event triggers similar tweets, the tweets are classified into ISSUE. For example, when a popular singer released a new song, we can often observe that people post tweets including the song title and media link. On the other hand, UNCLASSIFIED includes all tweets that we cannot properly classify, e.g., tweets in other language than English,

Table 9.7. Examples for near-duplicate types

Type	Examples
SPAM	weight loss support <i><link></i> / get a free estimate <i><link></i>
COMMON	lol / lets do this / whats up / hello new followers
ISSUE	watched mentalist [<i>TV series</i>] / coughing again

Table 9.8. Results of manual classification

SPAM	COMMON	ISSUE	UNCLASSIFIED
29%	37%	10%	24%

or tweets that we cannot infer the context of near-duplicates since the text in the tweets looks like a random string. Table 9.7 provides more examples for each type.

Table 9.8 shows the manual classification results. A large portion of near-duplicates is classified as SPAM or COMMON. When mining microblogs, we would not expect that these two types would contain meaningful information. On the other hand, if we can correctly identify ISSUE types, we would obtain helpful information which can be exploited for various tasks. For example, when we observe many near-duplicates of “Coughing again” in Table 9.7, we may use the fact to detect a disease outbreak.

In addition, we could not find any false positive cases, i.e., falsely detected pairs, in the 100 samples. This shows that the statistics of near-duplicates may be somewhat underestimated.

9.5 Conclusions

We reviewed a framework for text reuse detection, including various definitions and two fingerprinting techniques, i.e., DCT fingerprinting and LSH. By text reuse pattern analysis on two social applications, i.e., blogs and microblogs, using these algorithms, some interesting aspects of text reuse structures in social applications

were revealed. These results and algorithms can be used for text analysis studies such as information flow inference.

CHAPTER 10

CONCLUSIONS

In this thesis, in order to use social media applications as information sources, we focused on retrieval tasks and techniques exploiting three core structures beneficial for effective retrieval. Specifically, we discovered these structures and extracted relevant contexts from them. Using a geometry-based representation technique, we introduced how these contexts can be incorporated into retrieval frameworks. In addition, we discussed two more challenges that can serve as bases for promising future research, i.e., identification of relevant substructures and text reuse pattern analysis. To summarize all the discussions in our thesis, this chapter provides a brief review of each chapter. We then reiterate the contributions of our work. Finally, we propose multiple research directions for future work.

10.1 Summaries of Chapters

- **Chapter 3:** To justify a geometric mean-based representation framework used throughout this thesis, we described a generalized mean which can be defined on any metric space, i.e., Fréchet mean as a representation of multiple documents or contexts. Specifically, we assumed the Riemannian manifold based on the Fisher information metric and derived two approximated representations, i.e., the arithmetic mean and the geometric mean. Through empirical evaluation, the geometric mean is closer to the real representation and often leads to better retrieval performance for two generic IR tasks.

- **Chapter 4:** We defined hierarchical structures by ownership or containment relations, e.g., threads and postings. We focused on blog site search as a target task for exploring hierarchical structures blog site search. We introduced contexts which can be extracted from hierarchical structures and representation techniques based on these contexts. In experiments, local context-based blog site representations using the geometric mean-based approach suggested in Chapter 3 outperformed baselines. We showed that a diversity penalty factor based on global contexts is necessary to complement the geometric representations to improve performance.
- **Chapter 5:** We defined conversational structures by reply relations and addressed exploitation of these structures for online community search tasks. However, since reply relations are missing in many cases, we first introduced a thread structure discovery algorithm which demonstrated reasonable performance. Various local contexts were extracted from the predicted structures. Combinations of these contexts improved strong baselines. Further, in thread search, the techniques proposed for hierarchical structures in Chapter 4 worked well.
- **Chapter 6:** We defined social structures by authorship and introduced a novel graph-based expert finding technique to identify an influential role in the structures. This technique constructs a graph with structural relationships, e.g., reply relations and authorship, and performs a random walk. Using experiments on real online communities, we showed that integration of social structures as well as other structures can lead to a performance gain.
- **Chapter 7:** We summarized representation techniques introduced in Chapter 4, 5 and 6. Based on this summary, we proposed a unified framework combining three structures. For evaluation, we constructed a new test collection using

crowdsourcing. Experiments conducted on three forums demonstrated that we can benefit from this new framework.

- **Chapter 8:** It is not always easy to find specific relevant information from relevant retrieval objects which contain long text. We addressed identifying relevant substructures or subsets from these objects. With constraints from conversational structures, we proposed subset selection algorithms. These constraints led to better performance for identification of relevant substructures in forum threads.
- **Chapter 9:** We discussed text reuse structures implying relations among text fragments. We proposed text reuse detection algorithms for blogs and microblogs and analyzed text reuse patterns appearing in them. These tools and results can provide a basis for various research tasks such as information flow analysis and spam filtering in social applications.

10.2 Our Contributions

- **An understanding of unique structures in social media applications which imply social information and community knowledge**

We presented three core structures in social applications, i.e., hierarchical, conversational and social structures and discussed ways of defining each structure by components in the applications. We discussed how each structure reflects unique aspects of social applications. In addition, we addressed text reuse structures. By analyzing text reuse patterns in social applications, we can understand social applications better.

- **Algorithms for discovering explicit or implicit structures in social media applications and extracting useful contexts from the structures**
Social media structures are sometimes obscure or even missing. We provided

methods to discover these structures embedded in social applications. Also, we presented methods which extract proper granularity contexts from structures so that they can be exploited for representing relevant information for retrieval tasks.

- **Geometry-based representation model for multiple contexts**

We derived a novel way of representing multiple contexts using Information Geometry as a tool and showed that this technique can lead to better retrieval performance. Also, this interdisciplinary approach combining Information Retrieval and Information Geometry can provide a framework that may help developments of new IR approaches.

- **Retrieval models incorporating information extracted from social media structures to improve the effectiveness of search**

We proposed retrieval techniques for various structural evidence extracted from social media structures. Most of them are based on geometry-based representation models. These techniques demonstrated better performance than strong baselines. In addition, we showed that we can improve the presentation of search results by identifying relevant substructures in retrieval objects.

- **Evidence showing that social media structures can be helpful resources for utilizing social applications as information sources**

Most algorithms using structural evidence extracted from social media structures demonstrated superior performance than techniques not using structural evidence. In particular, when using all three core structures, we achieved the best performance.

- **Customization of retrieval models for various real-world applications**

We designed all tasks based on actual tasks of real-world applications. Accordingly, all test collections were built from real blogs, forums and microblogs. We

introduced customization approaches to apply our algorithm to each task and collection.

- **Practices for building test collections for social media search evaluation**

We made our own test collections for most of our tasks by hiring annotators or crowdsourcing. Descriptions about test collection building processes and experiences from them provide good guidelines for people planning to establish their own collections for social media applications.

10.3 Future Work

Although we addressed various issues related to search tasks in social media applications, some challenges still remain to be explored.

- **Development of Generalized Geometric Representations**

In this work, we mainly addressed structures in social applications. However, from a different perspective, another topic was to find effective representations which can be exploited for representing structures. Specifically, a geometric representation technique was derived from theoretical evidence based on Information Geometry. While this interdisciplinary approach combining Information Retrieval and Information Geometry has the potential to lead to a new line of Information Retrieval theory, our work is somewhat limited. For example, we derived all techniques from the language modeling framework. However, we may discover more general theorems which are also applicable to other retrieval frameworks such as BM25. Of course, to do this, we need to estimate parametric statistical models from such frameworks. Moreover, we considered only the geometric mean and the arithmetic mean for representations. Other methods, e.g., the harmonic mean, may be interpreted under the current theory. Also,

we should be able to understand models using multiple retrieval features, e.g., the dependency model, under our frameworks. Finally, it would be interesting to discover connections between these representations and other representations based on harmonic analyses such as topic models or factor models.

- **Consideration of Other Structures or Different Definitions**

We addressed hierarchical, conversational and social structures defined in a few specific ways. We may define these structures in different ways. For example, a hierarchical structure can be defined by hierarchical term clusters or document clusters, not by ownership. We may also consider more explicit social structures appearing in some applications, e.g., Facebook or microblogs, and address retrieval tasks using social networks. The “+1” feature of Google is an example of applications using social networks. In addition to our three core structures, we mentioned substructures in a set object and text reuse structures. Relevant substructures can be more explored for achieving better search result presentations. We may design text analysis based on information flows inferred from text reuse and ranking algorithms incorporating various features generated from text reuse structures. Also, there may be other important structures. For example, temporal structures play crucial roles in some applications, e.g., microblogs.

- **Retrieval Evaluation by Real Users**

We discussed how to make test collections of social applications for evaluation. In particular, we used two different approaches: by trained annotators and by crowdsourcing. However, in online communities such as forums, a topic is discussed in depth. Moreover, slang and dialects with which people outside the communities are not familiar are frequently used. That is, there are often cases in which even trained annotators can hardly understand conversations

appearing in a thread. If the task is expert identification, this problem would be even more serious. Even if we easily understand the contents of a forum, it is hard to determine which person is an expert without following various postings and threads for a certain amount of time. That is, making test collections for social applications is often more difficult than for ad hoc retrieval in that many social applications address more focused topics. Indeed, the best people capable of making reliable relevance judgments are themselves real users of the social applications. Therefore, an ideal case is that social applications are developed to receive feedback from actual users. For example, voting or rating mechanisms can be integrated into application designs or retrieval interfaces. To find effective ways to accomplish this is a promising research direction.

- **Novel Search Result Presentation**

As discussed in Chapter 8, a list of threads or a list of postings in response to a user query may not be an optimal result format for satisfying the user's information need. We suggested that a subset of postings whose length is reasonable may yield a better result. However, this is a preliminary study and needs to be investigated further. For example, the proper types of results may depend on the queries or applications. For example, if a query is a question, a list of answers might be the best format for presenting results. On the other hand, if the query intent is to research people's opinions about a topic, a set of postings containing diverse opinions or its summarization page including a graph presenting percentages of negative/positive opinions can be considered a desirable interface where an answer can be instantly acquired. Also, to demonstrate the effectiveness of these interfaces, comprehensive user studies may be required.

APPENDIX A

GEOMETRY OF MULTIPLE DOCUMENTS

A.1 Approximations to the Fréchet sample mean in the Riemannian manifold defined by the Fisher information metric

To find approximations to the Fréchet sample mean, we first consider the Kullback-Leibler (KL) divergence which is defined as follows:

$$\begin{aligned} D(\mathbf{x}||\mathbf{y}) &= \sum_{j=1}^{n+1} x^{(j)} \log \frac{x^{(j)}}{y^{(j)}} \\ &= \sum_{j=1}^{n+1} x^{(j)} (\log x^{(j)} - \log y^{(j)}) \end{aligned}$$

As $y \rightarrow x$, approximately by the Taylor expansion,

$$\log x^{(j)} - \log y^{(j)} \approx -\frac{(y^{(j)} - x^{(j)})}{x^{(j)}} + \frac{(y^{(j)} - x^{(j)})^2}{2(x^{(j)})^2} + O((y^{(j)} - x^{(j)})^3)$$

From this,

$$\begin{aligned} &D(\mathbf{x}||\mathbf{y}) + D(\mathbf{y}||\mathbf{x}) \\ &= \sum_{j=1}^{n+1} [x^{(j)} (\log x^{(j)} - \log y^{(j)}) + y^{(j)} (\log y^{(j)} - \log x^{(j)})] \\ &= \frac{1}{2} \sum_{j=1}^{n+1} \frac{(y^{(j)} - x^{(j)})^2}{x^{(j)}} + \frac{1}{2} \sum_{j=1}^{n+1} \frac{(x^{(j)} - y^{(j)})^2}{y^{(j)}} + O(\|\mathbf{y} - \mathbf{x}\|^3) \end{aligned} \quad (\text{A.1})$$

Since \mathbf{y} approaches \mathbf{x} along geodesic c linking them, we can parameterize the path by arclength s so that $c(s_0) = \mathbf{x}$, $c(s_1) = \mathbf{y}$ and $s_1 - s_0 = \text{dist}(\mathbf{x}, \mathbf{y})$. The difference

between two points is expressed by a product of the geodesic length and the tangent vector to the curve as follows:

$$y^{(j)} - x^{(j)} = (s_1 - s_0) \frac{\partial c^{(j)}}{\partial s} = \text{dist}(\mathbf{x}, \mathbf{y}) \frac{\partial c^{(j)}}{\partial s}$$

Then, the first term in Equation (A.1) can be rewritten as follows:

$$\begin{aligned} \frac{1}{2} \sum_{j=1}^{n+1} \frac{(y^{(j)} - x^{(j)})^2}{x^{(j)}} &= \frac{1}{2} \sum_{j=1}^{n+1} \frac{1}{x^{(j)}} \left(\text{dist}(\mathbf{x}, \mathbf{y}) \frac{\partial c^{(j)}}{\partial s} \right)^2 \\ &= \frac{1}{2} \text{dist}^2(\mathbf{x}, \mathbf{y}) \sum_{j=1}^{n+1} \frac{1}{c^{(j)}(s)} \left(\frac{\partial c^{(j)}}{\partial s} \right)^2 \\ &= \frac{1}{2} \text{dist}^2(\mathbf{x}, \mathbf{y}) \sum_{j=1}^{n+1} c^{(j)}(s) \left(\frac{\partial \log c^{(j)}}{\partial s} \right)^2 \\ &= \frac{1}{2} \text{dist}^2(\mathbf{x}, \mathbf{y}) I(s) \end{aligned}$$

where $I(s)$ is the Fisher information for s . By definition of the length of the curve,

$$\int_{s_0}^{s_1} I(s) ds = \text{dist}(\mathbf{x}, \mathbf{y}) = s_1 - s_0$$

Hence, $I(s) = 1$, and we finally have the following:

$$\frac{1}{2} \sum_{j=1}^{n+1} \frac{(y^{(j)} - x^{(j)})^2}{x^{(j)}} = \frac{1}{2} \text{dist}^2(\mathbf{x}, \mathbf{y}) \tag{A.2}$$

Similarly, the second term in Equation (A.1) can be also written as Equation (A.2). Therefore, we have an approximation of Equation (A.1) as follows:

$$\begin{aligned} D(\mathbf{x}|\mathbf{y}) + D(\mathbf{y}|\mathbf{x}) &= \text{dist}^2(\mathbf{x}, \mathbf{y}) + O(\|\mathbf{y} - \mathbf{x}\|^3) \\ &\approx \text{dist}^2(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Similar relationships between divergences and distances can be founded in various texts [3, 60].

From this approximation, we can express the Fréchet sample mean with the KL divergence as follows:

$$\bar{\Phi}(\mathbf{c}) \approx \sum_{i=1}^k (D(\mathbf{p}_i|\mathbf{c}) + D(\mathbf{c}|\mathbf{p}_i)) \hat{Q}(\mathbf{p}_i) \quad (\text{A.3})$$

This means that finding the Fréchet sample mean is reduced to finding the symmetrized Bregman centroid \mathbf{c}^F [91] which is defined as follows:

$$\mathbf{c}^F = \arg \min_{\mathbf{c}} \sum_{i=1}^k \frac{1}{2} (D_F(\mathbf{p}_i|\mathbf{c}) + D_F(\mathbf{c}|\mathbf{p}_i)) \hat{Q}(\mathbf{p}_i)$$

where $D_F(\mathbf{x}|\mathbf{y})$ is the Bregman divergence defined by $F(\mathbf{x}) - F(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla F(\mathbf{y}) \rangle$ and F is a generator function. For example, if F is the negative Shannon entropy, i.e. $\sum_j x^{(j)} \log x^{(j)}$, then the Bregman divergence is the same as the KL divergence. That is, the Bregman divergence is a generalized divergence. In addition, right-sided centroid \mathbf{c}_R^F and left-sided centroid \mathbf{c}_L^F are defined as follows:

$$\begin{aligned} \mathbf{c}_R^F &= \arg \min_{\mathbf{c}} \sum_{i=1}^k D_F(\mathbf{p}_i|\mathbf{c}) \hat{Q}(\mathbf{p}_i) \\ \mathbf{c}_L^F &= \arg \min_{\mathbf{c}} \sum_{i=1}^k D_F(\mathbf{c}|\mathbf{p}_i) \hat{Q}(\mathbf{p}_i) \end{aligned}$$

Nielsen and Nock [91] show that symmetrized Bregman centroid \mathbf{c}^F lies on a geodesic linking \mathbf{c}_R^F and \mathbf{c}_L^F via the Bregman Pythagoras' theorem. We can apply the result to the KL divergence.

To compute \mathbf{c}_R^F , we solve the following optimization problem.

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^k \hat{Q}(\mathbf{p}_i) \sum_{j=1}^{n+1} p_i^{(j)} \log \frac{p_i^{(j)}}{c^{(j)}} \\
& \text{subject to} && \sum_{j=1}^{n+1} c^{(j)} = 1 \\
& && c^{(j)} > 0 \quad \forall j
\end{aligned}$$

We can easily solve this using the method of Lagrange multipliers, and the solution coincides with the arithmetic mean as follows:

$$c_R^{F(j)} = \sum_{i=1}^k \hat{Q}(\mathbf{p}_i) p_i^{(j)}$$

For \mathbf{c}_L^F , we solve the following problem.

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^k \hat{Q}(\mathbf{p}_i) \sum_{j=1}^{n+1} c^{(j)} \log \frac{c^{(j)}}{p_i^{(j)}} \\
& \text{subject to} && \sum_{j=1}^{n+1} c^{(j)} = 1 \\
& && c^{(j)} > 0 \quad \forall j
\end{aligned}$$

Similarly, using the method of Lagrange multipliers, we compute \mathbf{c}_L^F as follows:

$$c_L^{F(j)} = \frac{\prod_{i=1}^k \left(p_i^{(j) \hat{Q}(\mathbf{p}_i)} \right)}{\sum_{j=1}^{n+1} \prod_{i=1}^k \left(p_i^{(j) \hat{Q}(\mathbf{p}_i)} \right)}$$

If $\hat{Q} = 1/k$, then this is the ordinary normalized geometric mean.

Therefore, the symmetrized Bregman centroid when F is the negative Shannon entropy, or the approximated Fréchet sample mean lies on the geodesic linking the arithmetic mean and the normalized geometric mean.

A.2 Visualization of document geometries

To show how multiple documents, the arithmetic mean and the normalized geometric mean are distributed in each geometry, we use the following visualization. First, we construct a weighted complete graph, where each node is a document or the mean and a weight is determined by a kernel reflecting each geometry.

For the Euclidean metric, we use the following heat kernel:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp \left(\left(- \sum_{j=1}^{n+1} (x_1^{(j)} - x_2^{(j)})^2 \right) / 4t \right)$$

where t is a time parameter.

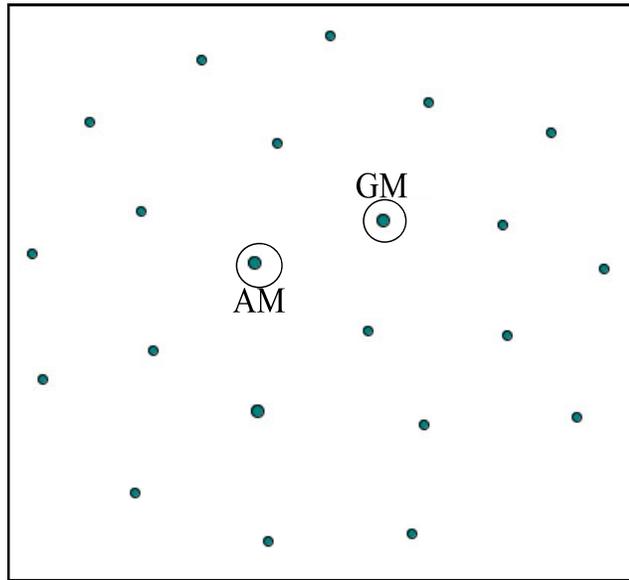
For the Fisher information metric, we use the following information diffusion kernel [68]:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp \left(- \arccos^2 \left(\sum_{j=1}^{n+1} \sqrt{x_1^{(j)} x_2^{(j)}} \right) / 4t \right)$$

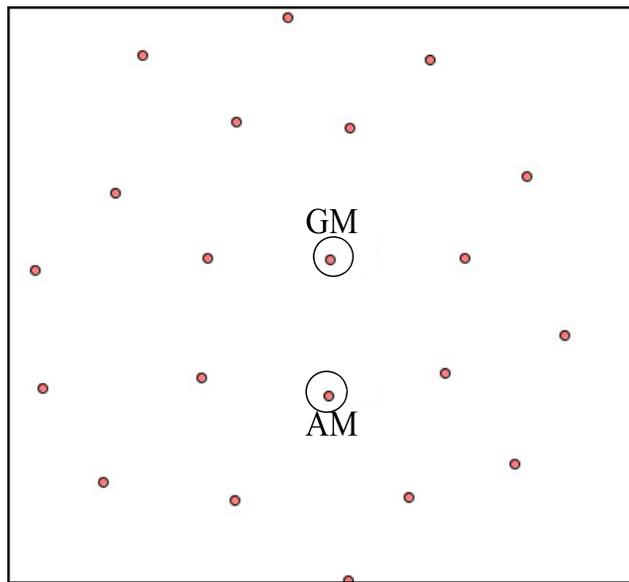
We visualize each geometry using CCVisu [10] which is a tool implementing energy models so that the higher weight between two points results in the smaller Euclidean distance between them. A visualization example is shown in Figure A.1. As you see, the arithmetic mean appears closer to the center in the Euclidean metric space while the normalized geometric mean appears closer in the Riemannian manifold defined by the Fisher information metric. Since the visualization tool uses random seeds to initialize the layout, the results vary every time. However, the trend for the locations of the means was consistent.

A.3 More accurate estimation for the approximated Fréchet sample mean

Geometric selection in Chapter 3 is a somewhat simple approach to determine the approximated Fréchet sample mean. That is, we choose one among only two options:



(a)



(b)

Figure A.1. Geometric visualization of the top 20 documents for Topic 770 (GOV2), the arithmetic mean (AM) and the normalized geometric mean (GM) for different metrics, i.e. the Euclidean metric (a) and the Fisher information metric (b).

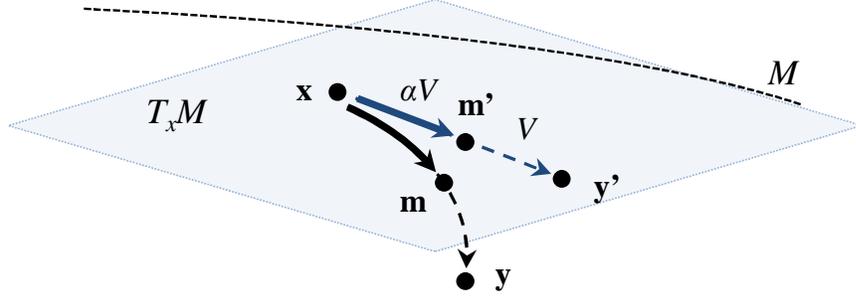


Figure A.2. Determination of a middle point \mathbf{m} on a geodesic linking \mathbf{x} and \mathbf{y}

the normalized geometric mean and the arithmetic mean. We now consider a more accurate estimation technique for the Fréchet sample mean.

A point which minimizes the approximated Fréchet sample function of Equation (A.3) lies on a geodesic linking the arithmetic mean and the normalized geometric mean. Let M , \mathbf{x} , \mathbf{y} and \mathbf{c} be the statistical manifold defined by the Fisher information metric, the arithmetic mean, the normalized geometric mean and a geodesic linking the two points, respectively. First, we get vector V on tangent space $T_{\mathbf{x}}M$ via log map $\log_{\mathbf{x}} : M \rightarrow T_{\mathbf{x}}M$. In case of a sphere, the log map is given by:

$$\begin{aligned} V^{(j)} &= \log_{\mathbf{x}}(\mathbf{y})^{(j)} \\ &= \frac{\arccos(\langle \mathbf{x}, \mathbf{y} \rangle)}{\sqrt{1 - \langle \mathbf{x}, \mathbf{y} \rangle^2}} (y^{(j)} - \langle \mathbf{x}, \mathbf{y} \rangle x^{(j)}) \end{aligned}$$

Then, V links \mathbf{x} to \mathbf{y}' on $T_{\mathbf{x}}M$ corresponding to \mathbf{y} on M .

\mathbf{m}' denotes a middle point between \mathbf{x} and \mathbf{y}' on $T_{\mathbf{x}}M$, reached by αV ($0 \leq \alpha \leq 1$). We now get a middle point \mathbf{m} on \mathbf{c} via exponential map $\exp_{\mathbf{x}} : T_{\mathbf{x}}M \rightarrow M$. The exponential map of a sphere is:

$$\begin{aligned} m^{(j)} &= \exp_{\mathbf{x}}(\alpha V)^{(j)} \\ &= \cos(\alpha \|V\|) + \frac{\sin(\alpha \|V\|)}{\|V\|} V^{(j)} \end{aligned}$$

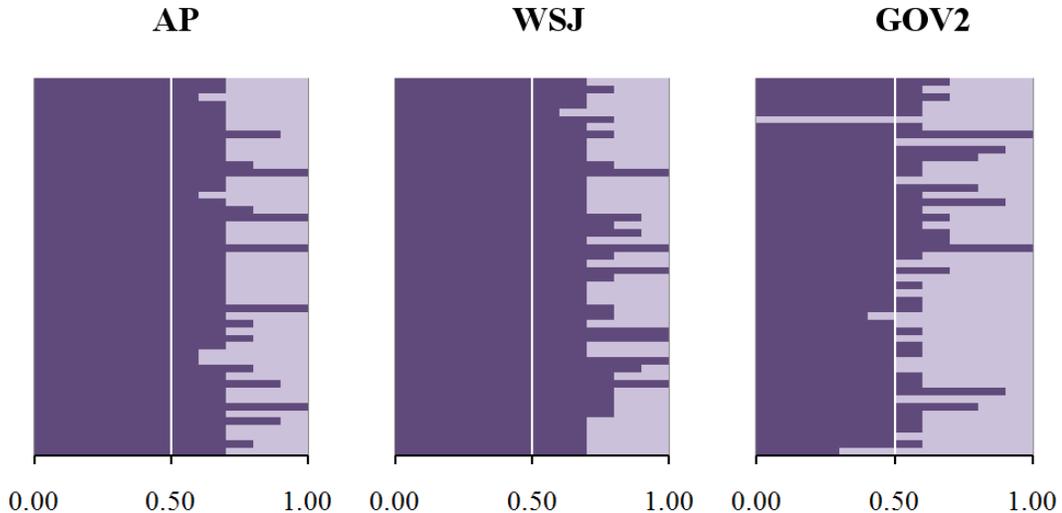


Figure A.3. Relative locations of the more accurately estimated Fréchet sample means. The x -axis corresponds to the relative locations, and the y -axis corresponds to queries for each collection. As a relative location is closer to 1.0, the estimated mean for the topic is located near the normalized geometric mean.

Table A.1. Pseudo-relevance feedback results of the more accurately estimated Fréchet sample mean in the Riemannian manifold defined by the Fisher information metric. GRM⁺ denotes the pseudo-relevance feedback technique using the more accurately estimated Fréchet sample mean. The results by RM and GRM are borrowed from Table 3.3.

	AP	WSJ	GOV2
RM	0.2541	0.3531	0.3204
GRM	0.2769*	0.3851*	0.3300*
GRM ⁺	0.2769	0.3852	0.3309

Figure A.2 illustrates this procedure. Note that the arithmetic mean \mathbf{x} and the geometric mean \mathbf{y} are interchangeable in the above formulation because a sphere is symmetric.

We apply this result to pseudo-relevance feedback experiments done in Chapter 3.4.2. We perform grid search on the geodesic varying α in $[0,1]$ by step-size 0.1, and a point which minimizes the Fréchet sample function of Equation (3.2) is selected as a representation. Figure A.3 shows α 's selected for test queries for each collection. For all test topics except for three topics of GOV2, the selected α 's are equal to or

greater than 0.5. That is, the more accurately estimated Fréchet sample means are also closer to the normalized geometric mean than the arithmetic mean. Table A.1 shows the results when the representations are used for pseudo-relevance feedback. All results are equal to or a little bit better than the results of the GRM, but not significantly. Therefore, we can say that the geometric relevance model is a reasonable approximation to the Fréchet sample mean for this task.

APPENDIX B

UNSUPERVISED ESTIMATION OF DIRICHLET SMOOTHING PARAMETERS

We often estimate language models of various social media structures or contexts using Dirichlet smoothing throughout this thesis. Dirichlet smoothing is known to be one of the most effective smoothing techniques for the language modeling-based retrieval framework [139]. This smoothing technique has a free parameter, i.e., the Dirichlet smoothing parameter. A standard approach for determining this parameter is to choose a value which maximizes a retrieval performance metric using relevance judgments. We call this supervised approach metric-based estimation of Dirichlet smoothing parameters.

We do not, however, always have relevance judgments as given by TREC standard test collections. For example, most of the collections used throughout this thesis are new collections that we crawled by ourselves. Therefore, there was no provided relevance judgment, and we should make our own relevance judgments via the pooling method [125]. Usually, the pooling method requires a number of initial runs to obtain ranked lists which contribute to the pool. For these initial runs, it would be advantageous to have a plausible Dirichlet smoothing parameters even though the parameter cannot be tuned by existing relevance judgments.

Also, even when we have relevance judgments for a collection, we may be addressing different search tasks from those for which relevance judgments are made. Furthermore, the characteristics of actual user queries can be different from the queries associated with relevance judgments used for training the smoothing parameter. For

example, if most queries used in relevance judgments are long, while real queries are short, then the trained value may not work well because the smoothing parameter is sensitive to query lengths as well as document lengths [82]. In fact, our own social media test collections have a small number of queries which may not be representative enough for all actual queries of social media applications. In such cases, it is not desirable to use metric-based estimation.

We introduce an unsupervised estimation approach which can be exploited under the circumstances. This method estimates a Dirichlet smoothing parameter from collection statistics, specifically, a variance of multinomial parameters associated with each term. Therefore, this estimation is independent of specific queries or relevance judgments. Note that if a test collection with relevance judgments is available, we cannot say that our unsupervised approach can produce a better smoothing parameter than the supervised approach. In this appendix, we intend to introduce an estimation technique which can be used when the supervised approach cannot be used.

There are few formal studies for determining Dirichlet smoothing parameters for retrieval models in an unsupervised manner. However, the average document length of a collection is sometimes used as the parameter value [37, 142, 97]. Also, in the Machine Learning literature, Minka [90] has presented maximum likelihood estimation for Dirichlet distributions.

B.1 Unsupervised Estimation

Dirichlet smoothing assumes that a document can be represented by a multinomial distribution, $\text{Multi}(\theta_1, \theta_2, \dots, \theta_N)$, where N is the size of vocabulary of collection C . Introducing a Dirichlet prior, $\text{Dir}(\alpha_1, \dots, \alpha_N)$, we choose the mean of the posterior distribution as a smoothed document representation given by

$$p(i|D) = \frac{tf_{i,D} + \alpha_i}{|D| + \alpha_0}$$

where D is a document, i is an index corresponding to a unique term, and

$$\alpha_0 = \sum_j \alpha_j$$

A typical choice for α 's is

$$\alpha_i = \mu \cdot m_i \quad \text{where} \quad m_i = \frac{cf_i}{|C|}$$

Then, the mean $E[\theta_i]$ and the variance $\text{Var}[\theta_i]$ of the Dirichlet prior are computed as follows:

$$\begin{aligned} E[\theta_i] &= \frac{\alpha_i}{\sum_j \alpha_j} = m_i \\ \text{Var}[\theta_i] &= \frac{[\alpha_i(\alpha_0 - \alpha_i)]}{[\alpha_0^2(\alpha_0 + 1)]} = \frac{m_i(1 - m_i)}{\mu + 1} \end{aligned}$$

While the mean is independent of μ , the variance is closely related to the choice of μ . Therefore, the variance can be parameterized by μ .

Assuming that a smoothing parameter should reflect collection statistics well, we choose μ which minimizes the following squared error of variances.

$$\begin{aligned} e(\mu) &= \sum_i \left(\frac{\bar{V}_i - \text{Var}[\theta_i]}{\text{Var}[\theta_i]} \right)^2 \\ &= \sum_i \left(\frac{\bar{V}_i(\mu + 1)}{m_i(1 - m_i)} - 1 \right)^2 \end{aligned}$$

where \bar{V}_i is the sample variance.

Via $\frac{de(\mu)}{d\mu} = 0$, a closed form solution is obtained by

$$\mu = \left(\sum_i \frac{\bar{V}_i}{m_i(1 - m_i)} \right) / \left(\sum_i \frac{\bar{V}_i^2}{m_i^2(1 - m_i)^2} \right) \quad (\text{B.1})$$

\bar{V}_i can be computed by

$$\bar{V}_i = \sum_{D \in \mathcal{C}} (p_{ML}(i|D) - m_i)^2$$

where $p_{ML}(i|D)$ is the maximum likelihood estimator of a language model, i.e., $tf_{i,D}/|D|$. However, since computations crossing all terms and all documents are required, this is practically infeasible in case of large collections. Therefore, we use a sampling and approximation approach. First, we randomly sample T terms from a collection and consider only these terms instead of all terms in vocabulary. Then, we exploit the fact that each term occurs very sparsely in documents. That is, in many cases, $tf_{i,D} = 0$. Accordingly, we consider the following approximation

$$\bar{V}_i \approx m_i^2$$

Using this approach, Equation (B.1) can be easily computed. We call this unsupervised approach variance-based estimation of Dirichlet smoothing parameters.

B.2 Empirical Evidence

We conducted experiments to evaluate our unsupervised estimation method. We used three standard TREC collections: AP (topic 51-150), WSJ (topic 51-150) and GOV2 (topic 701-800). Each document is stemmed by the Krovetz stemmer and stopped by a standard stopword list. To simulate situations where the characteristics of training queries are different from those of test queries, we split the topics into two subsets with the same size according to the number of terms in the topic titles, i.e., short queries and long queries.

For each collection, we considered four Dirichlet smoothing parameters. Two of them are values which maximize mean average precision (MAP) for short queries and long queries, respectively. To find the values, we swept [500, 4000] with stepsize 100. Another is the average document length of each collection that is often used as an

Table B.1. Average query lengths of split topic sets and four Dirichlet smoothing parameters. μ_{short} and μ_{long} are parameters trained for short queries and long queries, respectively. μ_{avgdl} is the average document length. μ_{est} is estimated by our proposed method.

	AP	WSJ	GOV2
Avg.#terms of short queries	2.5	2.5	2.4
Avg.#terms of long queries	5.1	5.1	3.8
μ_{short}	4000	2300	3700
μ_{long}	1900	1200	800
μ_{avgdl}	464	449	949
μ_{est}	2560	1563	1011

unsupervised heuristic for Dirichlet smoothing parameters. The last one is a value computed by our proposed method (with $T = 3000$). Table B.1 shows these values. As you see, even though relevance judgments are built on the same collection, there is a substantial divergence between the Dirichlet smoothing parameters trained for different types of queries. While the average document length does not appear close to the trained values, a parameter estimated by our unsupervised approach appears between two trained values. That is, this method seems to produce a plausible value.

We evaluated retrieval performance of these smoothing parameters for short queries and long queries. Table B.2 shows the results. The average document length produces consistently poor performance. Also, parameters trained with a specific type of query (μ_{short} and μ_{long}) do not generalize well to different types of queries. This shows that when making relevance judgments, accurate prediction of the characteristics of actual user queries is necessary so that the supervised approach is effective. On the other hand, parameters estimated by our unsupervised method, while not the best, do produce reasonable (i.e., the second best) performance regardless of the type of query for all collections.

To see how our method depends on the number of sample terms T , we tried various T 's as shown in Figure B.1. This shows that the Dirichlet smoothing parameter value

Table B.2. Retrieval results for short queries and long queries according to different Dirichlet smoothing parameters. A number is a MAP score.

	AP		WSJ		GOV2	
	Short	Long	Short	Long	Short	Long
μ_{short}	0.1359	0.1097	0.2255	0.1840	0.1532	0.1367
μ_{long}	0.1344	0.1114	0.2206	0.1853	0.1456	0.1479
μ_{avdl}	0.1304	0.1030	0.2107	0.1769	0.1466	0.1479
μ_{est}	0.1344	0.1109	0.2235	0.1847	0.1477	0.1477

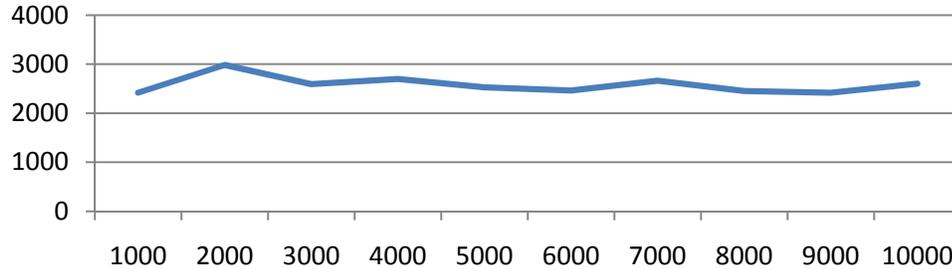


Figure B.1. Estimated Dirichlet smoothing parameters (y -axis) according to the numbers of sample terms (x -axis) on the AP collection.

appears stable after $T = 3000$. That is, the dependence on T is not substantial when a sufficient number of terms are used.

B.3 Conclusions

We introduced an unsupervised estimation approach for determining Dirichlet smoothing parameters. This method was shown empirically to be able to produce a plausible parameter. Furthermore, this method is relatively stable and robust in that it is independent of the characteristics of queries and relevance judgments. Therefore, it can be applied to cases that relevance judgments cannot be used or are not applicable as used for building test collections and conducting experiments in our social media search tasks.

BIBLIOGRAPHY

- [1] Allan, James. *Automatic Hypertext Construction*. PhD thesis, Cornell University, 1995.
- [2] Alonso, Omar, and Baeza-Yates, Ricardo. Design and implementation of relevance assessments using crowdsourcing. In *Advances in Information Retrieval* (2011), vol. 6611 of *Lecture Notes in Computer Science*, Springer, pp. 153–164.
- [3] Amari, Shunichi, and Nagaoka, Hiroshi. *Methods of Information Geometry*. American Mathematical Society, 2000.
- [4] Arguello, Jaime, Elsas, Jonathan, Callan, Jamie, and Carbonell, Jaime. Document representation and query expansion models for blog recommendation. In *Proceedings of the Second International Conference on Weblogs and Social Media (ICWSM 2008)* (2008).
- [5] Baeza-Yates, Ricardo, and Ribeiro-Neto, Berthier. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [6] Balog, Krisztian, Azzopardi, Leif, and de Rijke, Maarten. A language modeling framework for expert finding. *Inf. Process. Manage.* 45, 1 (2009).
- [7] Belkin, Nicholas J., Cool, C., Croft, W. Bruce, and Callan, James P. The effect multiple query representations on information retrieval system performance. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval* (1993), pp. 339–346.
- [8] Bendersky, Michael, and Kurland, Oren. Utilizing passage-based language models for document retrieval. In *Proceedings of the 30th European Conference on IR Research (ECIR 2008)* (2008), pp. 162–174.
- [9] Bernstein, Yaniv, and Zobel, Justin. Accurate discovery of co-derivative documents via duplicate text detection. *Information Systems* 31 (2006), 595–609.
- [10] Beyer, Dirk. CCVisu: Automatic visual software decomposition. In *Proc. Int'l Conf. on Software Engineering* (2008), pp. 967–968.
- [11] Bhattacharya, Rabi, and Patrangenaru, Vic. Nonparametric estimation of location and dispersion on riemannian manifolds. *Journal of Statistical Planning and Inference* 108 (2002), 23–35.

- [12] Bishop, Christopher M. Mixture models and EM. In *Pattern Recognition and Machine Learning*. Springer, 2006, pp. 423–459.
- [13] Brin, Sergey, Davis, James, and García-Molina, Héctor. Copy detection mechanisms for digital documents. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data* (1995), SIGMOD '95, pp. 398–409.
- [14] Broder, Andrei Z. On the resemblance and containment of documents. In *Proc. of the Compression and Complexity of Sequences* (1997), pp. 21–29.
- [15] Broder, Andrei Z., Glassman, Steven C., Manasse, Mark S., and Zweig, Geoffrey. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.* 29, 8-13 (1997), 1157–1166.
- [16] Buckley, Chris, Allan, James, and Salton, Gerald. Automatic routing and ad-hoc retrieval using SMART. In *The second Text REtrieval Conference (TREC-2) Proceedings* (1994).
- [17] Burges, Christopher J. C., Ragno, Robert, and Le, Quoc Viet. Learning to rank with nonsmooth cost functions. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems (NIPS 2006)* (2006), pp. 193–200.
- [18] Callan, James P. Passage-level evidence in document retrieval. In *SIGIR '94: Proceedings of the seventeenth annual international ACM SIGIR conference on Research and development in information retrieval* (1994), pp. 302–310.
- [19] Callan, Jamie. Distributed information retrieval. In *Advances in Information Retrieval*, W. Bruce Croft, Ed. Kluwer Academic Publishers, Norwell, MA, USA, 2000, pp. 127–150.
- [20] Campbell, Christopher S., Maglio, Paul P., Cozzi, Alex, and Dom, Byron. Expertise identification using email communications. In *CIKM '03: Proceedings of the twelfth ACM international conference on Information and knowledge management* (2003).
- [21] Carvalho, Vitor R., and Cohen, William W. On the collective classification of email "speech acts". In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 345–352.
- [22] Charikar, Moses S. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing* (2002), STOC '02, pp. 380–388.
- [23] Chentsov, N. N. *Statistical Decision Rules and Optimal Inference*. American Mathematical Society, 1982.

- [24] Chowdhury, Abdur, Frieder, Ophir, Grossman, David, and McCabe, Mary Catherine. Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst.* 20, 2 (2002), 171–191.
- [25] Collins-Thompson, Kevyn, and Callan, Jamie. Estimation and use of uncertainty in pseudo-relevance feedback. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), pp. 303–310.
- [26] Cong, Gao, Wang, Long, Lin, Chin-Yew, Song, Young-In, and Sun, Yueheng. Finding question-answer pairs from online forums. In *SIGIR '08: Proceedings of the 31th annual international ACM SIGIR conference on Research and development in information retrieval* (2008), pp. 467–474.
- [27] Cover, Thomas M., and Thomas, Joy A. *Elements of Information Theory*, second ed. Wiley-Interscience, Hoboken, NJ, USA, 2006.
- [28] Croft, W. Bruce, and Lafferty, John, Eds. *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.
- [29] Cronen-Townsend, Steve, Zhou, Yun, and Croft, W. Bruce. Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), pp. 299–306.
- [30] Efron, Bradley. Defining the curvature of a statistical problem. *The Annals of Statistics* 3, 6 (1975), 1189–1242.
- [31] El-Hamdouchi, Abdelmoula, and Willett, Peter. Comparison of hierarchic agglomerative clustering methods for document retrieval. *The Computer Journal* 32, 3 (1989), 220–227.
- [32] Elsas, Jonathan L., Arguello, Jaime, Callan, Jamie, and Carbonell, Jaime G. Retrieval and feedback models for blog distillation. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings* (2008).
- [33] Elsas, Jonathan L., Arguello, Jaime, Callan, Jamie, and Carbonell, Jaime G. Retrieval and feedback models for blog feed search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), pp. 347–354.
- [34] Elsas, Jonathan L., and Carbonell, Jaime G. It pays to be picky: an evaluation of thread retrieval in online forums. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (2009), pp. 714–715.

- [35] Elsner, Micha, and Charniak, Eugene. You talking to me? a corpus and algorithm for conversation disentanglement. In *the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technology Conference (ACL-08: HLT)* (2008), pp. 834–842.
- [36] Erera, Shai, and Carmel, David. Conversation detection in email systems. *Lecture Notes in Computer Science 4956* (2008), 498–505.
- [37] Fang, Hui, and Zhai, ChengXiang. An exploration of axiomatic approaches to information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 480–487.
- [38] Finn, Aidan, Kushmerick, Nicholas, and Smyth, Barry. Fact or fiction: Content classification for digital libraries. In *Joint DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries* (2001).
- [39] Fisher, Danyel, Smith, Marc, and Welser, Howard T. You are who you talk to: Detecting roles in usenet newsgroups. In *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences* (2006).
- [40] Fleiss, Joseph L., Levin, Bruce, and Paik, Myunghee Cho. *Statistical Methods for Rates and Proportions*, third ed. Wiley, New York, 2003.
- [41] Fox, Christopher J. A stop list for general text. *SIGIR Forum 24* (1990), 19–35.
- [42] Fox, Edward A., and Shaw, Joseph A. Combination of multiple searches. In *The second Text REtrieval Conference (TREC-2) Proceedings* (1994), pp. 243–252.
- [43] Fréchet, M. Les éléments aléatoires de nature quelconque dans un espace distancié. *Ann. Inst. H. Poincaré 10* (1948), 215–310.
- [44] Fu, Yupeng, Xiang, Rongjing, Liu, Yiqun, Zhang, Min, and Ma, Shaoping. Finding experts using social network analysis. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (2007), pp. 77–80.
- [45] Fuhr, Norbert, Kamps, Jaap, Lalmas, Mounia, Malik, Saadia, and Trotman, Andrew. Overview of the inx 2007 ad hoc track. In *Focused Access to XML Documents*, Norbert Fuhr, Jaap Kamps, Mounia Lalmas, and Andrew Trotman, Eds. Springer-Verlag, Berlin, Heidelberg, 2008, pp. 1–23.
- [46] Gillick, Dan, and Favre, Benoit. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing* (2009), ILP '09, pp. 10–18.
- [47] Gleave, Eric, Welser, Howard T., Lento, Thomas, and Smith, Marc. A conceptual and operational definition of ‘social role’ in online community. In *HICSS '09: Proceedings of the 42nd Hawaii International Conference on System Sciences* (2009).

- [48] Goldstein, Jade, Kantrowitz, Mark, Mittal, Vibhu, and Carbonell, Jaime. Summarizing text documents: sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), SIGIR '99, pp. 121–128.
- [49] Grady, Catherine, and Lease, Matthew. Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (2010), CSLDAMT '10, pp. 172–179.
- [50] Hearst, Marti A. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.* 23 (March 1997), 33–64.
- [51] Heintze, Nevin. Scalable document fingerprinting. In *1996 USENIX Workshop on Electronic Commerce* (1996).
- [52] Henzinger, Monika. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), SIGIR '06, pp. 284–291.
- [53] Hoad, Timothy C., and Zobel, Justin. Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.* 54, 3 (2003), 203–215.
- [54] Horowitz, Damon, and Kamvar, Sepandar D. The anatomy of a largescale social search engine. In *WWW '10: Proceedings of the 19th international conference on World Wide Web* (2010).
- [55] Järvelin, Kalervo, and Kekäläinen, Jaana. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (2000), pp. 41–48.
- [56] Jeffreys, Harold. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 186, 1007 (1946), 453–461.
- [57] Joachims, Thorsten. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (2002), pp. 133–142.
- [58] Jurczyk, Pawel, and Agichtein, Eugene. Discovering authorities in question answer communities by using link analysis. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), pp. 919–922.
- [59] Karcher, H. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics* 30, 5 (1977), 509–541.

- [60] Kass, Robert E., and Vos, Paul W. *Geometrical Foundations of Asymptotic Inference*. Wiley-Interscience, 1997.
- [61] Kaszkiel, Marcin, and Zobel, Justin. Passage retrieval revisited. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (1997), SIGIR '97, pp. 178–185.
- [62] Kendall, W.S. Probability, convexity, and harmonic maps with small image i: Uniqueness and fine existence. *Proc. London Math. Soc.* 61 (1990), 371–406.
- [63] Kleinberg, Jon M. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46, 5 (1999), 604–632.
- [64] Klimt, Bryan, and Yang, Yiming. Introducing the enron corpus. In *CEAS 2004 - First Conference on Email and Anti-Spam* (2004).
- [65] Kogan, J., Teboulle, M., and Nicholas, C. The entropic geometric means algorithm: An approach for building small clusters for large text datasets. In *the Workshop on Clustering Large Data Sets* (2003), pp. 63–71.
- [66] Krovetz, R. Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the sixteenth annual international ACM SIGIR conference on Research and development in information retrieval* (1993), pp. 191–202.
- [67] Kurland, Oren, and Lee, Lillian. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (2004), pp. 194–201.
- [68] Lafferty, John, and Lebanon, Guy. Diffusion kernels on statistical manifolds. *The Journal of Machine Learning Research* 6 (2005), 129–163.
- [69] Langville, Amy N., and Meyer, Carl D. Deeper inside PageRank. *Internet Mathematics* 1, 3 (2003).
- [70] Lappas, Theodoros, Liu, Kun, and Terzi, Evimaria. Finding a team of experts in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (2009), pp. 467–476.
- [71] Lavrenko, Victor, and Croft, W. Bruce. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), pp. 120–127.
- [72] Lebanon, Guy. *Riemannian Geometry and Statistical Machine Learning*. PhD thesis, Carnegie Mellon University, 2005.
- [73] Lee, Joon Ho. Analyses of multiple evidence combination. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (1997), pp. 267–276.

- [74] Leuski, Anton. Evaluating document clustering for interactive information retrieval. In *CIKM '01: Proceedings of the tenth ACM international conference on Information and knowledge management* (2001), pp. 41–48.
- [75] Lewis, David D., and Knowles, K. A. Threading electronic mail - a preliminary study. *Inf. Process. Manage.* 33, 2 (1997), 209–217.
- [76] Lin, Chen, Yang, Jiang-Ming, Cai, Rui, Wang, Xin-Jing, and Wang, Wei. Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (2009), pp. 131–138.
- [77] Lin, Jimmy, Abels, Eileen, Demner-Fushman, Dina, Oard, Douglas W., Wu, Philip, and Wu, Yejun. A menagerie of tracks at Maryland: HARD, Enterprise, QA, and Genomics, Oh My! In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings* (2007).
- [78] Liu, Xiaoyong, and Croft, W. Bruce. Passage retrieval based on language models. In *CIKM '02: Proceedings of the eleventh ACM international conference on Information and knowledge management* (2002), pp. 375–382.
- [79] Liu, Xiaoyong, and Croft, W. Bruce. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (2004), pp. 186–193.
- [80] Liu, Xiaoyong, and Croft, W. Bruce. Representing clusters for retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (2006), pp. 671–672.
- [81] Liu, Xiaoyong, and Croft, W. Bruce. Evaluating text representations for retrieval of the best group of documents. In *Proceedings of the 30th European Conference on IR Research (ECIR 2008)* (2008), pp. 454–462.
- [82] Losada, David E., and Azzopardi, Leif. An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval* 11, 2 (2008), 109–138.
- [83] Macdonald, Craig, and Ounis, Iadh. The TREC Blogs06 collection : Creating and analysing a blog test collection. Tech. Rep. TR-2006-224, University of Glasgow, Department of Computing Science, 2006.
- [84] Macdonald, Craig, Ounis, Iadh, and Soboroff, Ian. Overview of the TREC-2007 blog track. In *TREC 2007 Notebook* (2007).
- [85] Manber, Udi. Finding similar files in a large file system. In *Proc. of the USENIX Winter 1994 Tech. Conf.* (1994), pp. 1–10.

- [86] McCallum, Andrew, Corrada-Emmanuel, Andrés, and Wang, Xuerui. Topic and role discovery in social networks. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence* (2005), pp. 786–791.
- [87] McDonald, Ryan. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR research* (2007), ECIR'07, pp. 557–564.
- [88] Metzler, Donald, Bernstein, Yaniv, Croft, W. Bruce, Moffat, Alistair, and Zobel, Justin. Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (2005), CIKM '05, pp. 517–524.
- [89] Metzler, Donald, and Croft, W. Bruce. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 472–479.
- [90] Minka, Thomas. Estimating a Dirichlet distribution, 2003. <http://research.microsoft.com/minka/papers/dirichlet>.
- [91] Nielsen, Frank, and Nock, Richard. Sided and symmetrized Bregman centroids. *IEEE Transactions on Information Theory* 55, 6 (2009), 2882–2904.
- [92] Oard, Douglas, Elsayed, Tamer, Wang, Jianqiang, Wu, Yejun, Zhang, Pengyi, Abels, Eileen, Lin, Jimmy, and Soergel, Dagbert. TREC-2006 at Maryland: Blog, Enterprise, Legal and QA Tracks. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings* (2008).
- [93] Ogilvie, Paul, and Callan, Jamie. Hierarchical language models for retrieval of XML components. In *INitiative for the Evaluation of XML Retrieval (INEX) 2004* (2004).
- [94] Ounis, Iadh, de Rijke, Maarten, Macdonald, Craig, Mishne, Gilad, and Soboroff, Ian. Overview of the TREC-2006 blog track. In *The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings* (2007).
- [95] Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry. The PageRank citation ranking: Bringing order to the web. Tech. Rep. 1999-66, Stanford InfoLab, 1999.
- [96] Papadimitriou, Christos H., and Steiglitz, Kenneth. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.
- [97] Petkova, Desislava, and Croft, W. Bruce. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006)* (2006), pp. 599–608.

- [98] Petkova, Desislava, and Croft, W. Bruce. UMass at TREC 2006: Enterprise track. In *Text REtrieval Conference (TREC) 2006* (2007).
- [99] Petrović, Saša, Osborne, Miles, and Lavrenko, Victor. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), HLT '10, pp. 181–189.
- [100] Ponte, Jay M., and Croft, W. Bruce. Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries* (1997), pp. 113–125.
- [101] Porter, M.F. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.
- [102] Rabin, Michael O. Fingerprinting by random polynomials. Tech. rep., Harvard University, 1981. TR-15-81.
- [103] Rambow, Owen, Shrestha, Lokesh, Chen, John, and Lauridsen, Chirsty. Summarizing email threads. In *HLT-NAACL '04: Proceedings of HLT-NAACL 2004: Short Papers on XX* (2004), pp. 105–108.
- [104] Rao, C.R. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society* 37 (1945), 81–91.
- [105] Rivest, R. The MD5 Message-Digest Algorithm, RFC 1321, 1992.
- [106] Rocchio, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*, Gerard Salton, Ed. Prentice Hall, 1971.
- [107] Salton, Gerard, Allan, James, and Singhal, Amit. Automatic text decomposition and structuring. *Inf. Process. Manage.* 32, 2 (1996), 127–138.
- [108] Sanderson, Mark, Paramita, Monica Lestari, Clough, Paul, and Kanoulas, Evangelos. Do user preferences and evaluation measures line up? In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), SIGIR '10, pp. 555–562.
- [109] Schleimer, Saul, Wilkerson, Daniel S., and Aiken, Alex. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (2003), SIGMOD '03, pp. 76–85.
- [110] Seo, Jangwon, and Croft, W. Bruce. Blog site search using resource selection. In *CIKM '08: Proceedings of the seventeenth ACM international conference on Information and knowledge management* (2008), pp. 1053–1062.

- [111] Seo, Jangwon, and Croft, W. Bruce. Local text reuse detection. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), SIGIR '08, pp. 571–578.
- [112] Seo, Jangwon, and Croft, W. Bruce. Umass at TREC 2007 blog distillation task. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings* (2008).
- [113] Seo, Jangwon, and Croft, W. Bruce. Thread-based expert finding. In *the SIGIR 2009 Workshop on Search in Social Media (SSM 2009)* (2009).
- [114] Seo, Jangwon, and Croft, W. Bruce. Geometric representations for multiple documents. In *SIGIR '10: Proceedings of the 33th annual international ACM SIGIR conference on Research and development in information retrieval* (2010), pp. 302–310.
- [115] Seo, Jangwon, Croft, W. Bruce, and Smith, David A. Online community search using thread structure. In *CIKM '09: Proceedings of the eighteenth ACM international conference on Information and knowledge management* (2009), pp. 1907–1910.
- [116] Serdyukov, Pavel, Rode, Henning, and Hiemstra, Djoerd. Modeling multi-step relevance propagation for expert finding. In *CIKM '08: Proceedings of the seventeenth ACM international conference on Information and knowledge management* (2008).
- [117] Shivakumar, Narayanan, and García-Molina, Héctor. SCAM: A copy detection mechanism for digital documents. In *Proc. of the 2nd Ann. Conf. on the Theory and Practice of Digital Libraries* (1995).
- [118] Shivakumar, Narayanan, and García-Molina, Héctor. Finding near-replicas of documents on the web. In *Intl. Workshop on the World Wide Web and Databases* (1999).
- [119] Shrestha, Lokesh, and McKeown, Kathleen. Detection of question-answer pairs in email conversations. In *COLING '04: The 20th International Conference on Computational Linguistics* (2004).
- [120] Si, Luo, and Callan, Jamie. Unified utility maximization framework for resource selection. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management* (2004), pp. 32–41.
- [121] Smith, Marc, Cadiz, J. J., and Burkhalter, Byron. Conversation trees and threaded chats. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work* (2000), pp. 97–105.

- [122] Smucker, Mark D., Allan, James, and Carterette, Ben. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (2007), CIKM '07, pp. 623–632.
- [123] Snow, Rion, O'Connor, Brendan, Jurafsky, Daniel, and Ng, Andrew Y. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), EMNLP '08, pp. 254–263.
- [124] Soboroff, Ian, de Vries, Arjen P., and Craswell, Nick. Overview of the TREC 2006 enterprise track. In *Text REtrieval Conference (TREC) 2006* (2007).
- [125] Sparck Jones, Karen, and van Rijsbergen, Cornelis Joost. Information retrieval test collections. *Journal of Documentation* 32, 1 (1976), 59–75.
- [126] Strohman, Trevor, Metzler, Donald, Turtle, Howard, and Croft, W. Bruce. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis* (2005).
- [127] Veldhuis, R. The centroid of the symmetrical Kullback-Leibler distance. *IEEE Signal Processing Letters* 9, 3 (2002), 96–99.
- [128] Viégas, Fernanda Bertini. *Revealing individual and collective pasts: Visualizations of online social archives*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [129] Wanas, Nayer, El-Saban, Motaz, Ashour, Heba, and Ammar, Waleed. Automatic scoring of online discussion posts. In *WICOW '08: Proceeding of the 2nd ACM workshop on Information credibility on the web* (2008), pp. 19–26.
- [130] Wang, Lidan, and Oard, Douglas W. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2009), pp. 200–208.
- [131] Wang, Yi-Chia, Joshi, Mahesh, Cohen, William W., and Rose, Carolyn. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the Second International Conference on Weblogs and Social Media (ICWSM 2008)* (2008).
- [132] Welser, Howard T., Gleave, Eric, Barash, Vladimir, Smith, Marc, and Meckes, Jessica. Whither the experts? social affordances and the cultivation of experts in community Q&A systems. In *CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering* (2009), pp. 450–455.
- [133] Welser, Howard T., Gleave, Eric, Fisher, Danyel, and Smith, Marc. Visualizing the signatures of social roles in online discussion groups. *The Journal of Social Structure* 8 (2007).

- [134] Wu, Yejun, and Oard, Douglas W. Indexing emails and email threads for retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (2005), pp. 665–666.
- [135] Xu, Jinxi, and Callan, Jamie. Effective retrieval of distributed collections. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1998), pp. 254–261.
- [136] Xu, Jinxi, and Croft, W. Bruce. Topic-based language models for distributed retrieval. In *Advances in Information Retrieval*, W. Bruce Croft, Ed. Kluwer Academic Publishers, Norwell, MA, USA, 2000, pp. 151–172.
- [137] Xu, Jun, and Li, Hang. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (2007), SIGIR '07, pp. 391–398.
- [138] Yeh, Jen-Yuan, and Harnly, Aaron. Email thread reassembly using similarity matching. In *CEAS 2006 - Third Conference on Email and Anti-Spam* (2006).
- [139] Zhai, Chengxiang, and Lafferty, John. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), pp. 334–342.
- [140] Zhai, ChengXiang, and Lafferty, John. Two-stage language models for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (2002), pp. 49–56.
- [141] Zhang, Jun, Ackerman, Mark S., and Adamic, Lada. Expertise networks in online communities: structure and algorithms. In *The 16th International World Wide Web Conference (WWW '07)* (2007).
- [142] Zheng, Jianfeng, and Nie, Zaiqing. Language models for web object retrieval. In *Proceedings of the 2009 International Conference on New Trends in Information and Service Science (NISS '09)* (2009), pp. 282–287.