Exploring Topic Structure:

Coherence, Diversity and Relatedness

Jiyin He

Exploring Topic Structure:

Coherence, Diversity and Relatedness

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op woensdag 18 mei 2011, te 12:00 uur

door

Jiyin He

geboren te Hangzhou, China.

Promotiecommissie
Promotor: Prof. dr. M. de Rijke

Overige leden: Prof. dr. L. Hardman
Prof. dr. M. Lalmas
Dr. C. Monz
Prof. dr. A.P. de Vries

Faculteit der Natuurwetenschappen, Wiskunde en Informatica


SIKS Dissertation Series No. 2011-17.
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The investigations were supported by the Netherlands Organization for Scientific Research (NWO) under project number 612.066.512, by the E.U. IST program of the 6th FP for RTD under project MultiMATCH contract IST-033104 and by the PROMISE Network of Excellence co-funded by the 7th Framework Program of the European Commission, grant agreement no. 258191.

Cover design by Emmanuelle Beauxis-Aussalet
Printed and bound by Off Page, Amsterdam

ISBN: 978-94-90371-81-4

Acknowledgments

I am grateful to my advisor, Maarten de Rijke. Without him, this thesis would not have come to exist. With Maarten I learnt to write my first scientific paper, to survive my first deadline and to step into the field of information retrieval. Throughout all these years, the weekly meetings with him were inspiring and full of lively discussions. I especially appreciate that he has an open mind that allowed me to explore my own path in the field, while keeping me on the right track towards finishing my PhD.

Next, I would like to thank Martha Larson, my daily supervisor during the first year of my PhD, because of her never-ending faith in me, her critical thoughts, and the fact that she always made time for discussions even of little details.

Thanks to Lynda Hardman, Mounia Lalmas, Christof Monz, and Arjan de Vries for serving as my committee members.

I would also like to thank all my co-authors as they have contributed their ideas, experiments and texts to the materials in this thesis. In particular I would like to thank Merlijn Sevenster and his colleagues at Philips Research for our nice collaboration on the MedLink project; they have contributed various valuable input—including experimental data and annotations—, engaged in very useful discussions of ideas and frequently gave feedback on the work in my thesis.

Thanks to Bouke, Katja, Marc, Martha, Spyros and Valentin for the discussions and feedback on early versions of my thesis, and to Emma for designing the cover of the book.

Working at ILPS has been fun. I miss the daily coffee breaks, where we shared coffee, fresh air and inspiration ... and the numerous long nights with ILPSers, where we talked about free will, prime numbers, and the holy trinity.

Last but not least, I am grateful to my family and friends for their love and support for all these years. I am grateful to Marc, who was always by my side and supporting me during the days when I was writing this thesis.

Jiyin He
April 3, 2011
Amsterdam

Acronyms

AP Average Precision

ALG Automatic Link Generation

CLEF Cross-Language Evaluation Forum

FM-LDA Facet Model with Latent Dirichlet Allocation

HAC Hierarchical Agglomerative Clustering

HC Hierarchical Clustering

KNN K-Nearest Neighbor

IA-P Intent Aware Precision

IA-select Intent Aware select

IDF Inverse Document Frequency

INEX INitiative for the Evaluation of XML retrieval

IR Information Retrieval

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LM Language Model

MRR Mean Reciprocal Rank

MAP Mean Average Precision

MMR Maximum Marginal Relevance

NDCG Normalized Discounted Cumulative Gain

pLSA probabilistic Latent Semantic Analysis

RR Round-Robin

TF Term Frequency

TREC Text REtrieval Conference

UPGMA Unweighted Pair Group Method with Arithmetic mean

VSM Vector Space Model

Contents

Chapter 1

Introduction

Information Retrieval (IR) is "concerned with the structure, analysis, organization, storage, searching and retrieval of information" [207]. Concretely, an IR procedure typically consists of the following elements: a user who has an *information need* that is formulated as a *query*, a collection of *documents*,[1] and an IR system that aims to find documents from the collection that are *relevant* to the user's information need as expressed by the query.

The notion of *relevance*, as one may see from the procedure described above, is one of the most fundamental, if not *the* fundamental concept in IR [46]. However, the meaning of "relevance" has been under debate for decades, as is witnessed by the critical literature overview of Saracevic [217] in 1975 and his successive work 30 years later [216]. Following Schamber et al. [218], major views on different kinds of relevance can be roughly categorized as system-oriented relevance and user-oriented relevance. The former mainly concerns itself with the *topicality*, or *aboutness* of the retrieved results to the query, that is, whether the topic of a query matches the topic of a document. The latter is mainly concerned with the *usefulness* of the results to users.

The focus on topicality in relevance assessments is closely related to the start of the experimental evaluation of IR systems, known as the Cranfield experiments [43, 44]. In Cranfield I, queries were generated from a source document and the task was to retrieve the source document to obviate the need of explicit relevance judgement. In Cranfield II, queries were generated in the same way, but source documents were eliminated from the assessment and retrieved documents were manually assessed. Despite various criticisms [86, 237], the Cranfield experiments have become the paradigm for experimental evaluation of IR systems. One important contribution of the Cranfield experiments is the idea of creating re-usable test collections with fixed queries, document collection and relevance judgements, so that different systems can be compared in a fair and repeatable manner.

On the other hand, the user-oriented view of relevance argues that there is more to consider in an IR system than just topicality. Various notions of relevance are proposed,

[1] The notion of document encompasses items of any media type such as texts, images, or video clips.

such as psychological relevance [88] and situational relevance [107, 218], suggesting that relevance is a multidimensional concept that depends on both cognitive and situational factors. While appealing, experimental evaluation of these types of relevance can be difficult and expensive.

Mizzaro [181] classifies various notions of relevance in a four-dimensional space: (i) information source, such as documents or representations of documents, (ii) representation of the user's information need, (iii) time and (iv) components. The first two dimensions represent the typical interaction between documents and queries, concerning topical relevance. The third dimension suggests that a document not relevant to a query at a certain point in time, may be relevant to the same query later, or vice versa. The fourth dimension decomposes the first two dimensions into three components: (i) the *topic* that the user is interested in, (ii) the *task* of the user, i.e., the activity that the user will execute with the retrieved documents, and (iii) the *context* which *includes everything not pertaining to topic and task, but however affecting search taking place and the evaluation of results* [181].

In this thesis, we consider a number of IR tasks that concern the "fourth dimension" of relevance. Particularly, in these tasks, "topic structure" plays an important role in satisfying users' information need. Therefore, we take a unified perspective and explore approaches to those tasks with regards to the notions of topic and topic structure. Below, we start by introducing our notions of topic and topic structure. After that, in Section 1.2 we discuss the research themes we address in this thesis, which are built around the following three aspects of topic structure: coherence, diversity and relatedness. Then we summarize the contribution of the thesis in Section 1.3 and the organization of the rest of the chapters in Section 1.4. We close this chapter by describing the origins of the materials on which the thesis is based.

1.1 Topic and topic structure

Topic

The notion of "topic" refers to the representative theme or subject contained in a piece of text or in a cluster of texts that are semantically close to each other. The "text" we discuss here can be interpreted as a word, a phrase or a document.

Representations of topics

We identify two dimensions where topics may differ in their representations. First, topics can be represented in an *implicit* or an *explicit* way. An explicit representation involves assigning labels to a text which indicate the subject of the text. When using an implicit representation, the topic of a piece of text is indirectly expressed, for example, by means of a distribution of term frequencies. Second, topics can be represented *internally* or *externally*. An internal topic representation for a piece of text uses statistics or labels derived from the text itself, while an external representation represents the

	Implicit	Explicit
Internal	clusters, latent topics	summarizations, cluster-internal labels
External	relevance feedback	classification labels

Table 1.1: Examples of different types of topical representation.

subject of a text using external resources, for example entries from a thesaurus or a dictionary.

Both implicit representation and explicit representations can be internal or external. Table 1.1 lists a few examples of topic modeling methods that fall into one of the four categories. Here, we briefly discuss the examples in each category.

(i) *Internal and implicit representation.* When using clustering for topic modeling, topics can be represented by clusters of documents discussing similar themes or subjects; when using probabilistic topic modeling approaches such as Latent Dirichlet Allocation (LDA) [18], topics are defined as a set of latent variables that can generate terms that constitute a document according to a certain probability distribution. These are examples of internal and implicit representations, as no explicit labels are used, and no external resources are involved.

Note that while clusters and topic models can be used to represent topics, the use of clustering and topic models is not limited to discovering topics. For example, documents can be clustered with respect to authorship. It is the document representation, i.e., features used to describe a document, that determines whether documents are clustered together because they share similar topics. In this thesis, we make the assumption that clustering and topic models are used to discover topics and that the document representations we adopt aim to capture the topics discussed by the documents. This assumption is consistent with the assumption behind various cluster-based retrieval methods that will be discussed in Section 2.2.

(ii) *Internal and explicit representation.* Automatic summarization is an example of internal and explicit topic representation: it takes an information source, extracts content from it, and presents the most important content to the user in a condensed form [164]. The condensed output can be seen as the label of the original text extracted from the text itself. Another example is the so-called *cluster-internal labeling*: it produces labels for clusters so that users can see what a cluster is about and computes a label that solely depends on the cluster itself [167]. For example, labeling a cluster using the title of the document closest to the cluster centroid.

(iii) *External and explicit representation.* Classifying texts into predefined subject categories is a typical example of external and explicit topic representation. Here, the predefined subject categories explicitly indicate the topic of the texts assigned to them. The procedure of assigning texts to categories can be done manually as well as automatically, for instance using a machine learning technique.

(iv) *External and implicit representation.* The procedure known as query expansion with relevance feedback can be roughly described as extracting terms from a set of documents which are (assumed to be) relevant in order to enhance the original query

in formulating the information need (about a specific topic). These terms are generated from resources other than the query itself; often, the original query as well as the expanded query are represented using term statistics.

In this thesis, we choose to focus on two types of representation listed in Table 1.1: the internal and implicit representation and the external and explicit representation, as these two naturally fit into the scenario of the tasks we are going to address. See Section 1.2 for more details.

Topic structure

The notion of "topic structure" refers to a certain type of association present among topics. When examining topic structure in this thesis, we focus on three types of association: coherence, diversity and relatedness. In the next section, we formulate these as research themes with dedicated explanations for each theme and motivate our choices for this particular focus of the thesis.

1.2 Research themes

The general goal of the thesis is to analyze and exploit topic structure in the context of IR. Specifically, we identify the following three main research themes related to this general goal:

RT 1 Topical coherence: the degree to which a set of documents is focused on certain topic.

RT 2 Diversity and the cluster hypothesis: the relation between topical relevance, diversity and the cluster hypothesis and its implication for result diversification.

RT 3 Relating topics in different representations: linking terms from documents to their definitions in a knowledge base.

Next, we discuss these themes in a bit more detail.

RT1. Topical coherence

The first research theme we consider is *topical coherence*. Given a set of texts, the topical coherence of the set refers to the degree to which these texts are focused on certain topics. For example, given a set of documents, we are interested in questions such as: Do these documents focus on a single topic? Or do they focus on several different topics? Or are they just a set of documents with random topics?

In the part of the thesis that is devoted to RT1, we focus on the internal and implicit topical representation and topics are modeled via statistical approaches such as clustering. In this context, the above questions relate to the issue of determining the optimal number of clusters in clustering or the number of latent variables in "latent

topic" methods. While deciding on the optimal number of clusters has long been a difficult problem [20, 65, 89], solutions proposed to finding the number of latent topics are quite empirical [18, 55, 110] and computationally demanding. Moreover, although an analysis of the relation among topics can be performed once the clustering is done, the result is heavily dependent on the assumed number of clusters and on the clustering algorithm. Therefore, a measure independent of these factors is needed so that topical coherence can be measured in a consistent way.

In this thesis, we propose a coherence score that captures the topical coherence of a set of documents by measuring the relative tightness of its clustering structure as compared to a background collection. It is an implicit measure, that is, without explicitly conducting clustering or making assumptions about the number of optimal clusters. Within this context, the following research questions are addressed:

RQ1a. How do we measure the topical coherence of a set of documents?

RQ1b. Can the coherence score we propose effectively reflect the topical coherence of a set of documents?

We then apply the coherence score within the context of two retrieval tasks, namely blog feed retrieval and query performance prediction. The blog feed retrieval task is defined as *identifying blogs that show a central, recurring interest in a given topic*. We use the coherence score as a measure of the topical consistency among posts belonging to a given blog, and incorporate this measure in a language modeling based retrieval framework to blog feed retrieval. With respect to the blog feed retrieval task, we address the following three research questions:

RQ2a. How do we measure topical consistency for a blog?

RQ2b. How can we use the coherence score in our blog retrieval process?

RQ2c. How does the size of a blog influence the estimation of the coherence score of the blog and how does this influence blog feed retrieval?

A typical ad-hoc retrieval scenario is as follows: a user issues his or her information need in the form of a query and submits it to a retrieval system, and the system then aims to satisfy this information need by returning documents in a ranked list in descending order of their estimated topical relevance to the query.

For the query performance prediction task, we posit that in the ad-hoc retrieval setting, the level of ambiguity of a query is correlated with the retrieval performance for that query. In order to measure the ambiguity of a query, we measure the topical coherence of the set of documents associated with the words contained in a query and integrate them into query coherence scores as an indication of the query ambiguity. Given this scenario, we ask the following research questions:

RQ3a. Can we use the coherence score to measure query ambiguity?

RQ3b. Can we use query ambiguity as measured by coherence-based scores to predict query performance in an ad-hoc retrieval setting?

RT2. Diversity and the cluster hypothesis

One important hypothesis in IR is the cluster hypothesis [105, 121, 245], which states that *similar documents tend to be relevant to the same request*. Based on this hypothesis, many query-specific cluster-based retrieval approaches have been developed [50, 105, 121, 137, 140, 155, 263]. Query specific clustering is the idea of clustering retrieval results for a given query. The central assumption behind this type of approaches is that, given a query, relevant documents are more similar to each other than to non-relevant documents. These approaches have successfully improved the retrieval performance in the setting of ad-hoc retrieval, where the information need is satisfied as long as the top ranked documents are relevant.

In this thesis, we re-visit the cluster hypothesis in the context of result diversification, a scenario where the expectation of desired results for a retrieval system is different from that of ad-hoc retrieval. Specifically, in the setting of result diversification, the information need is satisfied when the top ranked documents are *relevant* and *diverse*. For example, if a query is ambiguous or multi-faceted, the top ranked documents are expected to cover all the relevant interpretations or facets of the query, while documents covering the same interpretation or facet are treated as redundant and undesirable. Given the above scenario, we ask the following research questions:

RQ4. How do we interpret the cluster hypothesis in the context of result diversification?

RQ5. Can query specific clustering be used to improve the effectiveness of result diversification?

RT3. Relating topics in different representations

So far we have been focusing on topic structure at the document level, or at the level of a set of documents and with an internal implicit topic representation. In the work devoted to the last research theme, we zoom in on the word and phrase level. Moreover, we turn to a different type of representation of topics, namely the explicit and external representation.

Substantial work has been done in modeling topics at the document level, where the language usage, i.e., statistics of the terms occurring in a document are used as representations of the underlying topics. While this type of implicit representation with term statistics has led to many successful statistical topic modeling methods, they all heavily rely on one assumption, that is, similar topics are expressed with similar language usage. Moreover, these methods usually require extensive contexts in order to be able to generate reliable statistics. This type of approach becomes problematic when the language usage is inconsistent. For example, in a medical document, "olfactory nerve" can be referred to as "1st cranial nerve" or simply "1st nerve." In this case, an explicit representation, such as a definition from a knowledge base, is useful as all these

different expressions can be mapped to the same unique concept in the knowledge base and therefore more robust statistics can be provided.

Automatically constructing mappings between terms and phrases found in free text and entries in a knowledge base is non-trivial. We study this problem in the context of Automatic Link Generation (ALG) with Wikipedia. That is, given a piece of text, we identify words or phrases whose meaning is important for understanding the whole text and we link these words or phrases to concepts in Wikipedia for explanations or background information. While not in the shape of a typical document retrieval task, ALG very clearly is a retrieval problem. The user's information need can be formulated as *find me background information from a knowledge base for important (domain specific) terms in the document I am currently reading*. The system then needs not only to link relevant information from the knowledge base for terms occurring in a document, but also needs to decide which terms should be linked: linking too many or too few terms can both lead to dissatisfaction of the user.

From the literature we see that existing ALG systems have shown satisfying performance on the related problem of generating links *between* Wikipedia pages [175, 178]. In this thesis, we aim to take a step further. First, we investigate the following research question:

RQ6 While exploring Wikipedia's link structure for relating two topical representations, what is the impact of the evaluation type, training collection and learning methods?

Further, we perform a case study where we automatically generate links for text data from the radiology domain to Wikipedia. We aim to answer the following research question through this case study:

RQ7 Can state-of-the-art ALG systems that are, in principle, domain independent, be effectively applied to linking texts from a specific domain to Wikipedia? If not, can we improve the effectiveness of automatic link generation by considering domain specific properties of the data?

1.3 Contributions

The main contributions of the thesis can be summarized as follows.

- We develop a coherence score that effectively measures the topical coherence of a set of documents.
- The coherence score is successfully applied to two IR tasks where a measure of topical coherence is needed, namely, blog feed retrieval and query performance prediction. In blog feed retrieval, our proposed approach effectively improves the retrieval performance. For query performance prediction, empirical results show that coherence-based query ambiguity scores are significantly correlated with the performance of queries as evaluated with a number of retrieval methods.

- We contribute to the understanding of the cluster hypothesis in IR by re-visiting the hypothesis in the context of result diversification, a scenario different from ad-hoc retrieval, in which it has typically been considered so far.

- We propose a cluster-based result diversification framework that effectively improves the performance of several existing result diversification methods. We provide an in-depth analysis of the relation between relevance, diversity and the cluster hypothesis within this framework.

- We study the problem of relating topics in different representations in the context of automatic link generation to Wikipedia. We analyze factors that impact the use of Wikipedia link structure for ALG, including evaluation types, training collections and learning methods. The result of the analysis provides implications for future work on this topic.

- We conduct a case study in the radiology domain where we automatically annotate radiology reports with background information from Wikipedia. Our study shows that in order to use ALG techniques on the data from the radiology domain, existing ALG systems trained on data from a general domain need non-trivial adaptations. On top of that, the ALG system we propose shows its effectiveness in linking medical concepts from radiology reports to Wikipedia concepts.

1.4 Organization of the thesis

The thesis is organized in ten chapters, grouped in three parts.

Chapter 2 This chapter provides background material for the work presented in this thesis. First, we briefly introduce basic concepts in IR, with an emphasis on topic representation and matching in different retrieval models. Some of the retrieval models are used in the rest of the chapters. Second, we survey the work that has been done in the cluster-based retrieval and discuss the cluster hypothesis, which is the basis of our work in Chapter 6 and 7. Then, we discuss related work on the retrieval tasks that we are going to address in each part of the thesis, where the notion of "relevance" is beyond "aboutness." At the end of the chapter, we specify the evaluation methodology employed in this thesis.

Chapter 3 In this chapter, we propose a coherence score that measures the topical coherence of a set of documents, and provide a theoretical analysis of some of the properties of the score and an empirical evaluation of the score on simulated data. We then empirically evaluate the effectiveness of our proposed coherence measure in two IR tasks in Chapter 4 and 5.

Chapter 4 Here, we evaluate the effectiveness of the coherence score in the context of blog feed retrieval, where the coherence score is used to measure the topical consistency among posts belonging to a single blog.

Chapter 5 Here, we evaluate the effectiveness of the coherence score in the context of query performance prediction. Specifically, the coherence score is used to measure the word ambiguity of a query as an indication for the difficulty of the query in retrieving relevant documents.

Chapter 6 In this chapter, we explore the impact of topic structure on effectively presenting retrieval results, with a focus on the scenario of result diversification.

Chapter 7 Inspired by the cluster hypothesis, we propose a result diversification framework based on query-specific clustering and cluster ranking. On top of that, we investigate the relation between relevance, diversity and the cluster hypothesis.

Chapter 8 Here, we study the problem of linking topics represented in different forms using automatic link generation techniques. We explore the impact of the following factors on machine learning based Automatic Link Generation approaches: evaluation type, training collection and learning approach.

Chapter 9 We present a case study of automatic link generation in the radiology domain, where we evaluate state-of-the-art link generation systems and propose our own approach that improves over the state-of-the-art systems on radiology data.

Chapter 10 This chapter concludes the thesis by re-visiting the research questions and reviewing our answers and contributions. On top of that, we discuss remaining open issues and future directions that follow up on the work of this thesis.

1.5 Origins

The work described in this thesis is based on the following publications. The coherence score in Part I was first introduced in [100], which is described in detail in Chapter 5. Chapter 4 is based on the work in [101] and its extension [102]. Part II of the thesis is mostly based on the work described in [104]. In Part III, the work on automatic link generation with Wikipedia described in Chapter 8 is built upon [97, 98] and the case study on radiology data in Chapter 9 is based on [103].

In addition, the work described in the following publications is closely related to the thesis; while not discussed in detail, it is incorporated at various points of the thesis: [94, 95, 96, 99, 146].

Chapter 2

Background

In this chapter, we provide background material for later chapters in this thesis. We start with an introduction to basic concepts in IR in Section 2.1, where we focus on topic representation and matching in ad-hoc retrieval. In Section 2.2 we take a closer look at a specific way to enhance topic representation and matching. We discuss the use of document clustering in IR, where the topic representation of a document and its matching against a query representation is enhanced by exploiting the topical structure present in the collection. Further, in Section 2.3, we discuss a number of retrieval tasks where the notion of "relevance" is beyond "aboutness." Moreover, in these tasks, *topic structure* plays an important role in satisfying a user's information need. Finally, in Section 2.4 we discuss the experimental evaluation methodology for IR systems that we use in later chapters.

2.1 Information retrieval

In a standard ad-hoc retrieval setting, the goal of a retrieval system is to find relevant documents that match a user's information need in a document collection, where an information need is understood to be the topic about which the user desires to know more, and a document is taken to be relevant if it is "about" the topic that the user is interested in [167]. In order to realize this goal, the following ingredients are necessary: (1) a representation of each of the documents that indicates the topics covered by the document; this is referred to as a *document model*; (2) a representation that expresses the user's information need; this is referred to as a *query model*; and (3) a *matching function* that matches the query model against document models and estimates the relevance of a document to the information need.

In the following subsections, we discuss these ingredients from the perspective of topic representation and matching. We separate the retrieval process in two stages: indexing and searching. In both stages, we focus on how topics covered by a document or requested by a query are captured and represented. In the searching stage, we also

discuss how matching functions use these representations to find documents covering the topics required by the user and represented by the query.

2.1.1 Indexing

The indexing process assigns *index terms* to documents and stores them in a way that allows efficient and effective access. These index terms constitute an *indexing language* that determines the vocabulary that is used to generate a document representation. Index terms may be derived from the text of the document to be described (internal), or they may be derived independently (external). Further, the index vocabulary can be *controlled* or *uncontrolled* [245].

A controlled vocabulary refers to a set of approved index terms, for example, a vocabulary derived from a manually maintained ontology or thesaurus. Indexing documents using a controlled vocabulary can be seen as assigning topic labels to documents from an external resource, that is, topical information is represented externally and explicitly. Early systems using a controlled vocabulary usually involved manual assignment of topical labels to documents, which is an expensive process. With the rapid growth in the volume of document collections that need to be searched, manual indexing with a controlled vocabulary was gradually replaced by automatic indexing with an uncontrolled vocabulary. Nevertheless, indexing with a controlled vocabulary is still useful in certain domains. A typical example retrieval system using a controlled vocabulary is the MEDLINE system for indexing and searching biomedical literature [145], which first became available in 1964 and is still in use today. In addition, attempts have been made to automatically map topics contained in a query and (or) documents to a thesaurus to enhance retrieval systems. For example, Giger [76] proposed to map both query and documents into a concept space in order to exploit the actual meaning of the information need and the documents. Voorhees [249] experimented with building an index that uses WordNet to disambiguate polysemous nouns and replaced those terms with their senses, which was shown to improve over a pure term-based index for some queries, although in general the term-based index was superior. Meij and de Rijke [172, 173] experimented with using thesaurus as a source for query reformulation.

Compared to indexing with a controlled vocabulary, automatic indexing with an uncontrolled vocabulary is cheap and efficient. Often, indexing terms are derived from the documents in the collection with certain word conflation, including (1) removal of high frequency words, (2) suffix stripping, (3) detecting equivalent stems [245]. While cheap and efficient, automatic indexing with a uncontrolled vocabulary was also proven to be effective [44, 209].

Two factors are considered important in choosing an index language, namely *specificity* and *exhaustivity*, where indexing exhaustivity is defined as the number of different topics indexed, and the index language specificity is the ability of the index language to describe topics precisely [132, 245]. Studies aimed at quantifying the two factors have been carried out, particularly, by associating them to document collection

statistics [159, 199, 210, 211, 234]. For example, exhaustivity can be assumed to be related to the number of index terms assigned to a given document, and specificity is assumed to be related to the number of documents to which a given term is assigned in a given collection. These statistics are closely related to term weighting schemes developed in different retrieval models.

Maron and Kuhns [168] were the first to propose probabilistic indexing for retrieval systems and suggested that there are two relationships between terms, namely, the semantic relationship that is based on the meanings of terms, and the statistical relationship that is based on the relative frequency of occurrence of terms used in an index. While the semantic relationships between terms are independent of the "facts" described by those terms, the statistical relationships are based on the nature of the facts described by the document. Indeed, it is the statistical relationship between terms that captures the topic discussed by the terms and therefore it is possible to implicitly represent topics covered by a document solely based on statistics of terms found in a document.

2.1.2 Searching

At search time, further representations of documents may be constructed, for example, by representing a document as a weighted term vector using term statistics derived from the index repository as weights. Further, the query needs to be represented in a way compatible to the document representation, so that matching is possible. Depending on the type of retrieval model, documents and queries are represented in different manners. Below, we discuss a number of representative retrieval models that differ in document and query representations as well as matching functions.

Boolean model

The Boolean model is the earliest retrieval model. Using the boolean model, the topics conveyed by a document or a query are represented by the presence or absence of index terms. Boolean operators such as AND, OR and NOT are used to match the query against documents. The documents returned by a boolean retrieval system form an (unranked) set. Under the boolean model, all terms are assumed to be equally important for the representation of a topic. Further, all documents that match the query are assumed to cover the requested topic to the same degree (if at all). Later, extended boolean models were proposed that introduced term weighting [69, 188, 212]. In spirit these models are very close to the vector space model (see below.)

Vector space model

In the Vector Space Model (VSM) [209], documents and queries are represented as term vectors in a high dimensional space, where each index term is an independent dimension of the space. If a term occurs in a document, it gets a non-zero weight in

the term vector of the document. The term weights can be binary, i.e., 0 for absence of a term and 1 for presence of a term, or real numbers, for example, the TF.IDF weights discussed below are commonly used.

To match the document representation and the query representation, a similarity score is calculated between the two term vectors. Cosine similarity is a frequently used similarity measure for term vectors with real values within the vector space model, which is defined as

$$cosine(\vec{d_1}, \vec{d_2}) = \frac{\vec{d_1} \cdot \vec{d_2}}{||\vec{d_1}|| \cdot ||\vec{d_2}||}, \tag{2.1}$$

where d_1 and d_2 are two documents represented in term vectors.

The VSM is a widely used model in IR and it is fundamental to a host of information retrieval operations ranging from scoring documents on a query to document classification and document clustering [167].

Probabilistic model

Probabilistic models are a set of retrieval models developed based on the *probabilistic ranking principle* (PRP) [198], which states that documents in a collection should be ranked in order of their probability of relevance to the query. The initial idea of probabilistic retrieval dates back to Maron and Kuhns [168]. Robertson [198] proved that the PRP is valid under certain assumptions, particularly, that the relevance of a document to a query is assumed to be independent from other documents.

Term weighting is an important theme in probabilistic models [235]. Terms are assumed to be associated with certain topics and a document may be about a topic or not. The term distribution over documents that are about a topic is assumed to be different from that over documents that are not about the topic.

Robertson and Spärck Jones [200] summarized three features to describe whether a term is a "good" one in terms of its ability to distinguish relevant documents from non-relevant ones, that is, a term that can characterize a topic and meanwhile discriminate it from other topics:

Collection frequency Terms that occur in only a few documents are often more valuable than ones that occur in many.

Within-document frequency The more often a term occurs in a document, the more likely it is to be important for that document.

Document length A term that occurs the same (absolute) number of times in a short document and in a long one is likely to be more valuable for the shorter document.

These features lead to the TF.IDF term weighting scheme. The basic TF.IDF term weighting schema can be described as follows. Let $D = \{d_i\}_{i=1}^{N}$ be a set of N documents, and $d = t_1, ..., t_m$ be a document with m terms. For a given term t and document

d, the term frequency (TF) of t is its within-document frequency with respect to d and the document length (DL) is the total number of words in d. The inverted term frequency (IDF) of t refers to its inverted document frequency [234] with respect to the collection D, which is defined as

$$IDF(t,D) = \log \frac{N}{df(t,D)}, \tag{2.2}$$

where $df(t,D)$ is the number of documents in D that contain t. A simple way of combining the three weights results in

$$TF.IDF = \frac{TF \cdot IDF}{DL}. \tag{2.3}$$

Many variations of each component, (i.e., TF, IDF, and DL) of the TF.IDF weighting have been developed, particularly in the context of probabilistic retrieval models [6, 87, 199, 201, 228]. For example, in the divergence from randomness (DFR) framework, a term is assume to be a "good" term if its within document frequency is higher than its expected frequency from a random distribution. In practice, this boils down to selecting a random distribution, which is the collection frequency, and applying two types of normalization of the within document frequency. Roelleke and Wang [205] studied the interpretation of TF.IDF with respect to various term weighting functions in different types of retrieval model such as the binary independence model [199], the two Poison model [201], the DFR [6] model, as well as the query likelihood language model [190].

Language models

A language model represents documents and queries with probability distributions over terms. These models originate from probabilistic models of language generation developed in the automatic speech recognition community [124]. Since the late 1990s, they have been successfully applied to information retrieval [16, 108, 176, 190].

Under the language modeling framework, each language model can be seen as an underlying topic that is expressed by the text. For a given text T with m terms $T = t_1, ..., t_m$, a language model defines a probability mechanism under which the text is generated. In the IR context, usually unigram models are used. That is, the occurrence of terms are assumed to be independent events. Based on the above assumptions, the probability of the text T is then defined as

$$p(t_1, t_2, \ldots, t_m|\theta_T) = \prod_{i=1}^{m} p(t_i|\theta_T). \tag{2.4}$$

Often, a multinomial distribution is assumed for θ_T, using a maximum likelihood estimation (MLE), the probability of a term t given θ_T is estimated as the relative frequency of t_i in T, formally:

$$p(t|\theta_T) = \frac{c(t,T)}{|T|}, \tag{2.5}$$

where $c(t,T)$ is the count of t occurring in T, and $|T|$ is the length of T.

Note that MLE is an inaccurate estimation based solely on observed data and this is especially true when T is a short text such as a query. In order to obtain a more robust estimation, the probability distribution $p(t|\theta_T)$ estimated from T is usually *smoothed* with a probability distribution derived from a background model θ_B that is often constructed from a large collection of documents with a sufficiently large amount of terms to provide a reliable prior probability of a term occurring in a text. Jelinek-Mercer smoothing [125] is a popular and conceptually simple smoothing technique, where $p(t|\theta_T)$ is estimated as a linear interpolation between the model θ_T and the background model θ_B:

$$p(t|\theta_T) = (1-\lambda)p(t|\theta_T) + \lambda\, p(t|\theta_B), \tag{2.6}$$

where the parameter α is used to control the amount of smoothing. Many smoothing techniques exist; Zhai and Lafferty [268] have studied the role of smoothing in language models and empirically compared the impact of a number of popular smoothing techniques on retrieval effectiveness.

Various matching functions were proposed to estimate the relevance for a query of a document within a language modeling framework. The original method is referred to as the *query likelihood* model, where the relevance of a document given a query is interpreted as the probability of a query $Q = q_1, ..., q_n$ derived by a document model θ_d. Using the Bayes rule, the query likelihood is calculated as

$$p(\theta_d|Q) = \frac{p(Q|\theta_d)p(\theta_d)}{p(Q)} \propto \prod_{i=1}^{n} p(q_i|\theta_d) \tag{2.7}$$

Since the goal is usually to rank a set of documents according to the query likelihood score with respect to a query, the normalization term $p(Q)$ is a constant and can therefore be dropped for convenience. Further, the prior probability $p(\theta_d)$ is often assumed to follow a uniform distribution for simplicity.

An alternative matching approach is to measure the (dis)similarity between two language models, for example, between a query model and a document model. The *Kullback-Leibler (KL) divergence* is a measure often used to compare two language models [143, 262]. Using the KL divergence, the similarity between two language models, e.g., a query model θ_q and a document model θ_d is estimated as follows:

$$KL(\theta_q||\theta_d) = \sum_{t\in V} p(t|\theta_q) \log \frac{p(t|\theta_q)}{p(t|\theta_d)}, \tag{2.8}$$

where V is the vocabulary of all terms over which the language models are built.

KL divergence is not only used as a matching function for a query and a document, but also as a distance measure in other applications such as clustering [141, 262]. For example, Xu and Croft [262] proposed to use KL divergence in two settings. In a retrieval setting, it is used to measure how well a topic model (i.e., document language model) predicts a query; and in a clustering setting, it is used to estimate the closeness of a document to a cluster.

Topic models

A number of topic models have been proposed in the literature that aim at capturing the underlying "latent" topics from observed documents. Here we briefly discuss three representative models, including Latent Semantic Analysis (LSA) [55], probabilistic Latent Semantic Analysis (pLSA) [110] and Latent Dirichlet Allocation (LDA) [18].

LSA uses a vector space model representation of documents. By applying a singular value decomposition (SVD) on a co-occurrence matrix of terms and documents, it constructs a lower rank matrix where each component represents a latent topic. By mapping terms or documents to these latent topic components, terms or documents that share similar topics are grouped together.

The pLSA follows roughly the same idea as LSA but a probabilistic interpretation. The basic idea is that a term is generated as a mixture of latent topics, and a term t is conditionally independent from a document d given a latent topic z:

$$p(t,d) = p(d)\sum_{z} p(t|z)p(z|d). \tag{2.9}$$

Blei et al. [18] pointed out that the formulation of pLSA is not a well defined generative model as it learns the topic mixtures $p(z|d)$ only for those documents on which it is trained on and therefore there is no natural way to use it to assign probabilities to unseen documents. This problem is addressed in the LDA model.

The basic idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The generative process of a document of n words can be described as follows.

> Choose a latent variable $\theta \sim Dirichlet(\alpha)$
>
> For each of the n words w_i:
>
> > Choose a topic $z_i \sim Multinomial(\theta)$;
> >
> > Choose a word w_i from $p(w_i|z_i,\beta)$, a multinomial probability conditioned on the topic z_i.

Given a training set of documents, the model parameters can be estimated using variational inference with the expectation-maximization (EM) algorithm [18]. An alternative inference technique uses Gibbs sampling [80].

2.1.3 Summary

In this section, we have discussed topic representations and matching functions commonly used in ad-hoc retrieval. The general goal is to capture the topics discussed by a document and match these against the topic that a user is interested in. A topic representation consists of two key elements: the index terms and the logical representation of the index terms, as characterized by the retrieval models.

In the rest of the thesis, we will occasionally use some of the retrieval models or document representations. For example, when calculating the coherence score (see Chapter 3), we use the VSM with TF.IDF weighting to represent documents, and use cosine similarity as the similarity measure. In Chapter 4 our basic retrieval model for blog feed search uses a language model with a query likelihood matching function. In Chapter 7 we conduct clustering using two types of topic representation. We use the VSM for hierarchical clustering and LDA to model the underlying topics covered by a document.

2.2 The cluster hypothesis and cluster-based retrieval

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [120]. Jain et al. [120] have provided a survey on clustering techniques from a statistical pattern recognition perspective, and more recently Berkhin [17] completed a survey with an emphasis on data mining problems with large data sets and complicated attribute types.

The use of document clustering in information retrieval has been studied for decades. An early review on hierarchical document clustering for IR is provided by Willett [260], and a relatively recent review can be found in [167]. Recently Carpineto et al. [34] have conducted a survey on Web clustering engines. Among the many studies in cluster-based retrieval, some aim to improve retrieval performance in terms of effectiveness [50, 105, 121, 137, 138, 139, 140, 155, 156, 157, 158, 241, 251, 263], others aim to improve retrieval efficiency [4, 29, 30, 31, 49, 208] or both [30, 31, 229].

2.2.1 The cluster hypothesis

The central assumption behind the idea of using clustering to enhance retrieval effectiveness is the *cluster hypothesis*. In the literature, the cluster hypothesis has been formulated in different but closely related ways. An early and widely adopted version is formulated as "closely associated documents tend to be relevant to the same requests" [121, 245]. A formulation that focuses more on the distribution of document similarities between relevant and non-relevant documents is "relevant documents tend to be more similar to each other than to non-relevant documents" [105, 245].

As pointed out by van Rijsbergen [245], the assumptions made by the cluster hypothesis can only be verified by experimental work on a large number of collections. In addition, it also depends on how the hypothesis is tested [61, 79, 251]. Some early work has shown positive results in examining the validity of the hypothesis [105, 121, 246]. In this thesis, we posit that the validity of the cluster hypothesis should be verified not only against different collections but also against different types of queries, since the relevance of a document is determined with respect to specific queries. In Chapter 6 we re-visit the cluster hypothesis with respect to a specific type of queries, namely ambiguous and multi-faceted queries.

2.2.2 Cluster-based retrieval

Early work in cluster-based retrieval typically uses clusters created at the collection level [50, 79, 121, 251] and hierarchical clustering [83] methods are preferred to partition-based [83] clustering methods. The use of partition-based clustering methods is mainly motivated by a concern for efficiency [54, 207, 227, 267], while the retrieval effectiveness using partition-based clustering is proven to be inferior to that of a traditional document based retrieval [209]. In the hierarchical clustering setting, both top-down and bottom-up search techniques were used to search the clusters in response to a query along the hierarchy [50, 121, 251], where the latter was shown to be more effective [50, 60]. In many studies, only a single cluster was retrieved for a query and the cluster was retrieved in its entirety [48, 50, 121]. Voorhees [251] shows that retrieval of entire clusters in response to a query usually results in poorer performance than retrieval of individual documents from clusters. Griffiths et al. [79] suggested that more than one cluster should be retrieved, e.g., either the 5 top-ranked clusters were retrieved or a sufficient number of clusters were retrieved to give a total of 10 distinct documents. Among these studies, there is no conclusive evidence that cluster-based retrieval can improve the retrieval effectiveness compared to document-based retrieval.

More recently, document clustering has been combined with the language modeling framework [11, 141, 155, 258]. These models have shown improved retrieval effectiveness compared to standard language models. In most cases, soft clustering methods were used: Azzopardi [11] and Wei and Croft [258] used LDA for topic modeling and Kurland and Lee [141] used K-Nearest Neighbor (KNN) [83] to generate overlapping clusters.

Apart from query independent clustering, *query-specific clustering*, an approach that clusters search results in response to a given query, has been shown to effectively improve search result quality [105, 137, 140, 241]. Preece [191] was one of the first researchers to propose the use of clustering to analyze search results. Willett [261] examined the effectiveness of query specific hierarchic clustering for IR. The query specific clustering strategy was found to be more efficient than query independent clustering as only relatively small subsets of a collection need to be clustered, while the effectiveness of a query specific method is not substantially inferior to that of the query independent method. However, it was suspected that the work of Willett [261] has limitations in the clustering algorithm as well as in the approach used to select documents to be clustered [241].

Hearst and Pedersen [105]'s work was the first to show that query-specific clustering can improve the retrieval effectiveness. As illustrated by Hearst and Pedersen [105], with a proper clustering algorithm, one can generate clusters such that a large percentage of the relevant documents retrieved are contained in a few *high quality* clusters. If we would be able to identify those clusters for a given query and place the documents they contain at the top of the ranking, retrieval performance can be substantially improved in terms of early precision. Later, Tombros et al. [241] carried out

a comparative study to examine the effectiveness of query specific clustering for IR, over multiple collections and multiple clustering algorithms. His results provided further motivation for the application of hierarchic query-specific clustering to IR based on improved effectiveness.

More recently, Kurland extensively studied methods to rank query specific clusters under a language modeling framework [138, 140, 142]. On top of that, re-ranking search results using query-specific clusters as a means of smoothing the document language model [139, 239] or for query expansion [149] were also shown to be able to improve retrieval effectiveness.

While all the query-specific clustering based retrieval methods discussed above aim to improve ad-hoc retrieval effectiveness as measured using standard precision and recall-based metrics, in Chapter 7 we explore the merits of query-specific clustering for result diversification, where the top ranked documents are expected to be both relevant to the query and covering diverse aspects of the query (see below).

2.3 Beyond "aboutness"

In the previous sections, we have discussed topic representation and matching in ad-hoc retrieval where the notion of relevance is defined as topical relevance or "aboutness." With the introduction of evaluation conferences such as the Text REtrieval Conference (TREC) [252], came a renewed focus on topical relevance in the IR community. These evaluation conferences continue the experimental evaluation tradition set up by the Cranfield experiments. Meanwhile, certain aspects from the "fourth" dimension described by Mizzaro [181] (see Chapter 1) are also addressed. In particular, in this thesis, we shall discuss a number of retrieval tasks where the notion of "relevance" is beyond the "aboutness." In these tasks, *topic structure* plays an important role in satisfying user's information need.

The first task we discuss here is the blog distillation task defined in the TREC Blog Track [162], in which the "task" component of the "fourth" dimension is addressed: a blog feed is judged to be relevant if the posts in that blog show a central, recurring interest in a given topic. Here, in order to be considered as relevant, a blog should not only mention information that is "about" the topic requested by the query, but also contain a dominant amount of information "about" the topic.

The second task is the diversity task in the Web Track [40], where the "context" component is addressed in the following way: top ranked documents are not only topically relevant to a query but also cover diverse aspects of a query; previously seen or known information is considered as redundant and undesired. Here, the relevance of a document to a query is not only determined by its own "aboutness" of a certain topic, but also of other documents that have been retrieved.

Another task we are going to introduce is called ALG, which is defined as: identify significant terms in a source text, and link these terms to entries in a knowledge base in order to provide background information. On the one hand, the goal is to en-

hance the topic representation of the source text by external resources. On the other hand, the linking procedure requires matching between two topic representations. In this scenario, the "aboutness" can be seen as part of the information need where the background information is "about" the identified significant term. On top of that, one needs to identify the topic of the source text as well as the (main) topic conveyed by the term in order to determine the terms to be linked with as well as the target entry in the knowledge base to be linked to.

2.3.1 Blog distillation

In response to the growing interest in blogs and methods to access blog content, the Text REtrieval Conference (TREC) launched a Blog Track in 2006 [187]. The first year this track ran, its main focus was on identifying relevant and opinionated blog *posts* given a topic. Since the launch of this track, many new insights into blog post retrieval have been gained [162, 179, 187]. TREC 2007 introduced a new task in the Blog Track: blog (or feed) distillation [162] (in this thesis referred to as blog feed retrieval). The aim is to return a ranking of blogs, rather than individual posts, given a topic; this is summarized as *find me a blog with a central, recurring interest in a given topic*. The scenario underlying this task is that of a user searching for feeds of blogs about a particular topic to add to a feed reader. This task is different from a filtering task [197] in which a user issues a repeating search on posts, constructing a feed from the results.

The main difference between the approaches applied by the different sites participating in TREC is the indexing unit used in the retrieval system: either full blogs [63, 223], or individual posts [63, 64, 223]. On top of either index, techniques like query expansion using Wikipedia [63] or topic maps [150] are applied. Seki et al. [221] proposed to capture the recurrence patterns of a blog using the notions of time and relevance. After an initial retrieval run on a blog index, the relevance of all posts in the blog is determined and plotted against time. The area underneath this plot is considered to reflect the recurring interest of this blog for the given topic. Some additional techniques proved to be useful (e.g., query expansion), but most approaches did not lead to significant improvements over a baseline, or even led to a decrease in performance.

A number of studies are aimed at modeling topical noise in blogs in order to capture the central/recurring interest of a blog in a topic. The voting-model-based approach of [160] is competitive with the TREC-2007 blog feed search results reported in [162]. Their approach identifies three possible topical patterns and formulates models that attempt to encode each of them into the blog retrieval model. As in [255], *central interest* is captured using a query-based cluster score designed to reflect the relevance of the central topic of the blog to the query. *Recurring interest* is captured using a query-based date score that breaks the temporal window of the data collection down into time-based intervals and sums a topical contribution from each interval. Tuning involves setting the optimal width of the time based interval. This approach resembles the one taken in [64], which incorporates topical relevance from the most recent

interval rather than from all intervals. *Central and recurring interest* is captured by the integration of a score measuring the cohesiveness of the language models used in the set of posts in a blog. Seo and Croft [222] use a range of "diversity factors" to measure the topical noise of a blog and penalize blogs with a topically diverse set of posts. The penalty is then integrated into their retrieval model which formulates the blog feed search problem as a resource selection problem, that is, select the best resource (collection of posts) for a given query. In Chapter 4 we use the coherence score to encode the topical structure of blogs, which allows us to simultaneously capture the topical focusedness at the blog level and the relatedness of sub-topics within the blog.

Apart from the attempts to model topical noise, various authors have experimented with ways to improve the retrieval effectiveness in blog feed search, including (i) index pruning [223, 257], e.g., removing blogs with a single post that are very unlikely to demonstrate recurring interest in a topic; (ii) exploiting various blog specific features such as comments and recency, as an indication of the importance of a post to its parent blog [256, 257]; and (iii) mixing different representations of blog posts [257] (e.g., combining a title representation with a content representation).

2.3.2 Result diversification

Diversification of search results has been recognized by many as an important issue [25, 77]. Zhai et al. [270] argue that it is insufficient to simply return a set of relevant documents where relevance of a document is treated independently from other retrieved documents, an observation that gives rise to new evaluation metrics and retrieval strategies that consider dependence among documents. Chen and Karger [37] investigate a scenario where the user is satisfied with a limited number of relevant documents instead of all relevant documents. They show that in such a scenario, it is more effective to optimize the expected value of a given metric and to rank documents in such a way that the probability of finding at least a relevant document among the top N is maximized. On top of that, they find that explicitly aiming to find only one relevant document inherently promotes diversity of documents at the top of a ranked list.

An early diversification method is Maximal Marginal Relevance (MMR) in which the merit of a document in the ranked list is computed as a linear combination of its similarity to the query and the smallest similarity to documents already returned [32]. Zhai and Lafferty [269] propose a risk minimization framework in which loss functions are defined according to different assumptions about relevance so as to minimize the user's average "unhappiness." A probabilistic version of MMR is proposed within this framework, a mixture model of novelty and relevance. Carterette and Chandar [35] propose a probabilistic facet retrieval model for diversification, with the assumption that users are interested in all facets that are potentially related to the query and thus all hypothesized facets are equally important.

Radlinski et al. [192] propose a method that learns a diverse ranking of retrieval results from users' clicks. Yue and Joachims [266] study a learning algorithm based on structural SVM that identifies diverse subsets in a given set of documents.

Agrawal et al. [1] propose a diversification method, *IA-select*, that uses the Open Directory Project to model facets associated with queries and documents. Unlike previous work in modeling underlying facets of a query such as the probabilistic facet model [35], IA-select takes into account the importance of individual user intentions.

Recently, Santos et al. [215] explore query reformulation for result diversification. Similar to IA-select, during the diversification procedure, the merit of a single document is estimated based on its relevance to the query, its coverage of the query aspects and its novelty to other retrieved documents. The difference is that underlying facets associated with a query are uncovered in the form of sub-queries.

In Chapter 7, we tackle the problem of result diversification using a query-specific approach based on cluster ranking.

2.3.3 Automatic link generation

Automatically generating links has a long history, going back well over a decade. Early publications include [2, 19, 62, 78]. Later commercial approaches have met with limited success [115, 183]. In the context of Wikipedia, renewed interest in automatic link generation emerged. A relatively early paper on the topic is [67], where the problem of discovering missing links in Wikipedia is addressed. The proposed method consists of two steps: first, clustering highly similar pages around a given page, and then identifying candidate links from those similar pages that might be missing on the given page. The main innovation is in the algorithm that is used for identifying similar pages and not so much in the link detection. Meanwhile, the task of disambiguating links to Wikipedia has received special attention as part of semantically oriented tasks such as named entity normalization in recent years. Cucerzan [53] uses automatically generated links to Wikipedia to disambiguate named entities in news corpora. Generalizing Cucerzan [53]'s work to user generated content with additional heuristics, Jijkoun et al. [126] focus on the named entity normalization task on blogs and comments. Recently, Meij et al. [173] study the problem in the scenario of semantic query suggestions, where each query is linked to a list of concepts from DBpedia, ranked by their relevance to the query.

The work that is closest to our work discussed in this thesis (Chapter 8 and 9) was presented in [175, 178]. The Wikify! system reported in [175] implements a two-stage process for link generation, namely, keyword extraction followed by word sense disambiguation, which corresponds to anchor text identification and target page finding, respectively. Particularly, for keyword extraction, the system experimented with TF.IDF and χ^2 statistics that characterize the importance of the terms in a document. Their most successful approach is the so-called *keyphraseness* measure, which is the likelihood of a phrase being an anchor text based on the observation of the existing links. For target identification, the Wikify! system employs a knowledge-based approach combined with a data-driven approach with part-of-speech features, using the disagreement between the two approaches as a measure to filter out unreliable links. Milne and Witten [178] tackle the same problem with machine learning techniques

and, in particular, contextual information in the source text was used to determine target pages, which in turn also served as features for anchor text detection. Their approach greatly improved the performance in terms of precision and recall.

The basic evaluation of both systems is done through automatic assessments, i.e., using existing Wikipedia links as ground truth and evaluating the performance of the systems in re-generating existing Wikipedia links. On top of that, Mihalcea and Csomai [175] conduct a Turing test to compare the performance of human annotators and their system on a set of randomly selected Wikipedia pages. Milne and Witten [178] conduct a manual assessment of the performance of their system on a news collection.

In Chapter 9 we will further discuss the two systems. We use them as baseline systems and compare their performance against our proposed link generation system in automatically generating links form radiology reports to Wikipedia, where the radiology reports are manually annotated with links to Wikipedia. We will discuss in detail the pros and cons of the systems when applied to data from a specific domain such as the radiology domain.

In 2007, INEX (the INitiative for the Evaluation of XML retrieval) launched the Link-the-Wiki (LTW) Track, which uses the Wikipedia collection as its test set, where the automatic link generation task is treated as a ranking problem. That is, both anchor texts and linked target pages are presented as a ranked list, ordered by relevance to the topic page. Automatic assessment with Wikipedia ground truth as well as human assessments are employed at INEX. One important issue discovered through human assessments is that there exist many trivial links in Wikipedia which are actively rejected by human assessors [113]. In fact, when one evaluates the Wikipedia ground truth against the manual assessments, the performance of Wikipedia ground truth is far from perfect.

In the LTW Track, link generation is evaluated at different levels, including: (i) file-to-file level, (ii) anchor-to-BEP (best entry point) level, and (iii) anchor-to-file level. At the file-to-file level, the evaluation procedure only considers whether a link should exist between two files, while where to start a link (i.e., the identification of an anchor text) is not considered. At the anchor-to-BEP level, not only the anchor text where a link starts is considered but also where the link points to in the target file, i.e., the best entry point in the target file, is considered. Evaluation at the anchor-to-file level is the same as the anchor-to-BEP, where BEP is set to 0, i.e., the start point of a file. This is the same as the automatic link generation task we consider in this thesis.

Within the setting of the LTW Track, various heuristics exploiting the statistics of existing Wikipedia links as well as retrieval-based methods have been proposed [111, 112, 114]. Machine learning based approaches were investigated but with limited success [130]. In [98], we have focused on a subtask of the link generation problem, namely, the target finding task, within a learning-to-rank framework. In Chapter 8 we further evaluate a number of factors that may have an impact on the performance of machine learning based approaches to automatic link generation with Wikipedia; these approaches aim to combine various heuristics in a systematic fashion.

2.4 Experimental evaluation of IR systems

In this section, we briefly introduce evaluation methodologies widely adopted in the IR community and employed throughout this thesis. Then we discuss a number of commonly used measures for evaluating system effectiveness that will be used later in our experiments, followed by a discussion on significance testing for the evaluation results.

2.4.1 Evaluation methodology

Evaluation is an important theme in research on information retrieval. Robertson [196] has provided a discussion on the long history of evaluation experiments in IR and the impact of those early experiments on today's practice of experimental evaluation of IR systems. The earliest experimentation dates back to the Cranfield experiments in the 1960s [43, 44]. One of the significant achievements of the Cranfield experiments was to define the methodology of IR experimentation [196]. One of the most important traditions set up by the Cranfield experiments is the employment of standard test collections. A test collection consists of (i) a fixed document collection, (ii) a fixed set of queries representing users' information need, and (iii) a set of relevance judgements that indicate whether a document is relevant to a given query. Such test collections enable fair and repeatable comparisons between systems and repetition of experimental results. Recently, Sanderson [213] has surveyed the methods and practice of research conducted in the evaluation of IR systems under this framework.

The Cranfield paradigm has later been adopted and enhanced by TREC [252] and other evaluation conferences that focus on information retrieval, for example, the INitiative for the Evaluation of XML retrieval (INEX) that focuses on XML retrieval, and the Cross-Language Evaluation Forum (CLEF) that has an emphasis on cross-lingual retrieval. Within each of the evaluation conferences, different tracks are created, which are often defined based on the nature of data collections or search tasks, for example, the Web Track focuses on searching in collection of Web pages, the Genomic Track focuses on searching in biomedical literature, etc. Within each track, a number of specific retrieval tasks are defined. For example, in 2009 the Web Track included two tasks: an ad-hoc retrieval task and a result diversification task.

One major difference between the TREC evaluation (and other evaluation conferences) and that of the early experiments are the documents to be judged for relevance [196]. Given the increased sizes of document collections adopted at TREC, it has become intractable to have exhaustive relevance judgement as in the early experiments, and therefore relevance judgements have to be selective. A commonly used strategy is the pooling method. That is, for each query, a document pool is created by selecting top ranked documents returned by a range of different retrieval systems and judged for relevance. Zobel [274] has shown that results based on the relevance judgements formed from a limited depth pool are reliable – if the pool is sufficiently deep – both for systems that contributed to the pool and for "new" systems.

2.4.2 Evaluation measures

Below, we introduce the evaluation measures that are frequently used in IR experiments and later in this thesis.

Typically, to calculate an evaluation score, we need two input variables: the retrieved documents in response to a query and their corresponding relevance judgements. With respect to the input of retrieved documents, the measures discussed here can be roughly categorized into set-based measures and rank-based measures. For a set-based measure, the order of retrieved documents under evaluation does not affect the scores. A rank-based measure takes into account the order of the documents. Further, with respect to the input of the relevance judgements, some measures accept binary judgements, i.e., a document is judged as either "relevant" or "non-relevant" with respect to a query, others accept graded judgements, i.e., a document is judged to be relevant to a query at different levels. In Table 2.1 we list the evaluation measures discussed in this section, along with their properties.

measure	**set-based**	**rank-based**	**binary**	**graded**
precision/recall/F-measure	x		x	
precision@X	x		x	
reciprocal rank		x	x	
average precision		x	x	
normalized discounted cumulative gain		x		x
α-NDCG		x		x
intent aware precision@X		x	x	

Table 2.1: A summary of evaluation measures discussed in this section and their properties.

The above evaluation measures are calculated over a set/ranked list of documents retrieved in response to a single query. In order to obtain a stable evaluation of the performance of a retrieval system, these scores are averaged over a set of test queries. In the case of reciprocal rank and average precision, the averaged evaluation results are referred to as Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP), respectively. For the rest of the measures listed here, conventionally, no change is made to their titles when averaging is performed.

Precision, recall and F-measure

Precision and *recall* are some of the earliest measures used for the effectiveness of retrieval systems, which dates back to the Cranfield II experiments [44]. Simply put, precision is the fraction of retrieved documents that are relevant; and recall is the fraction of relevant documents that are retrieved [167].

For a set of documents retrieved by an IR system and a set of binary relevance judgements, (i.e., each document is judged as either "relevant" or "non-relevant" with respect to a query), a contingency table can be constructed as in Table 2.2:

	Relevant	Non-relevant
Retrieved	tp (true positive)	fp (false positive)
Not retrieved	tn (true negative)	fn (false negative)

Table 2.2: A contingency table

Then, the score of precision is calculated as

$$P = \frac{tp}{tp+fp}, \tag{2.10}$$

and recall is calculated as

$$R = \frac{tp}{tp+fn}. \tag{2.11}$$

Both precision and recall are set-based measures and use binary relevance judgement. Often, a precision-recall curve can be used to visualize the retrieval performance of a ranked list. The precision-recall curve plots the precision value at different recall levels to show the trade-off between the two scores. See Fig. 8.1 on page 145 for an example.

Precision and recall can be summarized into a single score by using the *F-measure*, which is the weighted harmonic mean of precision and recall [167].

$$F_\beta = \frac{(1+\beta^2)P \cdot R}{\beta^2 P + R}. \tag{2.12}$$

The above general form of F-measure is derived by van Rijsbergen [245]: *it measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision.* When $\beta = 1$, a balanced F-measure that equally weights the precision and recall is derived.

Precision@X (P@X)

For a ranked list of documents retrieved in response to a query, the P@X score is the precision score at rank X. Let $rel(d) = 0$ when d is judged as non-relevant, and $rel(d) = 1$ when it is judged as relevant. The P@X score is calculated as:

$$P@X = \frac{1}{X}\sum_{i=1}^{X} rel(d_i). \tag{2.13}$$

Average precision

Average Precision (AP) combines precision and recall in a way that ranking relevant documents higher in a ranked list is favored. It is the average of the precision scores obtained at the ranks of relevant documents in a ranked list. For a ranked list of documents $D = d_1, ..., d_m$, the AP score is defined as:

$$AP(D) = \frac{1}{R}\sum_{i=1}^{m} P@i \cdot rel(d_i), \tag{2.14}$$

where R is the total number of relevant documents found in the collection.

AP (MAP) is one of the most widely used evaluation measures in the TREC community [167]. Buckley and Voorhees [28] have shown that for general purpose retrieval, AP is a reasonably stable and discriminating choice. Recently, Robertson et al. [204] proposed to extend AP to use graded relevance judgement.

Reciprocal rank

The reciprocal rank is defined as the reciprocal of the first retrieved relevant document. If no relevant document is retrieved, the reciprocal rank is defined as 0.

$$ReciprocalRank(D) = \frac{1}{r}, \tag{2.15}$$

where r is the rank where the first relevant document is found in the ranked list D. The reciprocal rank is a suitable measure for retrieval effectiveness when the users are interested in seeing a relevant document as early as possible in a ranked list. Recently, Chapelle et al. [36] proposed expected reciprocal rank, which can be seen as an extension of the classical reciprocal rank to the graded relevance case.

Normalized discounted cumulative gain

The Normalized Discounted Cumulative Gain (NDCG) score proposed by Järvelin and Kekäläinen [122] is a rank-based score and is designed to reflect graded relevance judgement. Given a ranked list of documents $D = d_1, ..., d_m$, a corresponding gain vector G is defined where $G[i]$ is the relevance judgement of the document at position i, for example, 0 for non-relevant, 1 for relevant and 2 for highly relevant, etc. Then a cumulative gain vector is defined as follows

$$CG[i] = \sum_{j=1}^{i} G[j]. \tag{2.16}$$

Further, the discounted cumulative gain is defined such that documents with high relevance but ranked low in the ranked list receive a discount factor. Many different discount functions exist, for example, Järvelin and Kekäläinen [122] define it as $\log_b j$ where $b \leq j$. Here, we follow Clarke et al. [41] and define the discounted cumulative gain as

$$DCG[i] = \sum_{j=1}^{i} G[j]/\log_2(1+j). \tag{2.17}$$

Finally, the discounted cumulative gain is normalized against the ideal cumulative gain, which is calculated using Eq. 2.17 over the ranked list of documents sorted by their judged relevance to the query:

$$NDCG[i] = \frac{DCG[i]}{DCG'[i]}. \tag{2.18}$$

α-NDCG

Based on NDCG, Clarke et al. [41] proposed the α-NDCG measure that aims to measure the effectiveness of result diversification. The goal of the diversity task is to return a ranked list of documents that together provide complete coverage for a query, while avoiding excessive redundancy in the result list [40]. The α-NDCG measure is employed at TREC 2009 and TREC 2010 Web Track as a measure for the diversity task [40, 42].

The major difference between the α-NDCG and NDCG is the way the cumulative gain is calculated. Assume a query has m subtopics, and $J(d,i)$ is the relevance judgement of document d with respect to subtopic i. For α-NDCG, the cumulative gain is defined as

$$CG[k] = \sum_{i=1}^{m} J(d_k, i)(1-\alpha)^{r_i,k-1}, \tag{2.19}$$

where

$$r_i, k-1 = \sum_{j=1}^{k-1} J(d_j, i) \tag{2.20}$$

is the number of documents ranked before d_k that are relevant to subtopic i; α can be interpreted as if a subtopic is covered by a document ranked before k, the probability that the user is still interested in a document that is relevant to the same subtopic.

Intent aware precision

The Intent Aware Precision (IA-P)@X is another measure used at TREC 2009 Web Track for result diversity [40]. It is adapted based on the intent aware measures proposed by Agrawal et al. [1]. Let N be the number of subtopics associated with query q. Let $j_q(i,j) = 1$ if the document returned for query q at depth j is judged relevant to subtopic i of query q; otherwise, let $j_q(i,j) = 0$. Then IA-P at retrieval depth X is defined as:

$$IA-P@X = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{X}\sum_{j=1}^{X} j_q(i,j). \tag{2.21}$$

Evaluation measures used in this thesis

We choose different evaluation measures for different tasks.

In Chapter 5 we use AP as an indication of system performance in a general purpose adhoc retrieval setting.

In Chapter 4 we use MAP, MRR, and P@X for measuring blog feed search effectiveness. In Chapter 7 we use α-NDCG and IA-P@X for measuring the effectiveness of result diversification. Note that in Chapter 1 we have briefly mentioned that for the blog feed search task and the result diversification task that is discussed in Chapter 6 and 7, the notion of relevance is beyond topicality or "aboutness." In addition to topicality, the blog feed search task requires that the retrieved blogs show a central and

recurring interest on the topic issued by the query, and the diversity task requires that the retrieved documents cover as many facets of the query as possible. Here, for result diversification, we use the measures specifically designed for this task, while for blog feed search, we simply use the measures used for adhoc retrieval systems. This is because the relevance judgements of the test collections we use for the two tasks are made in different ways. Unlike the result diversification task, the relevance judgements of the blog feed search task take into account the requirement in addition to topicality, and therefore adhoc measures can be directly applied.

Further, in Chapter 8 and 9 we use the precision and recall measures and their combination to evaluate the performance of automatic link generation. In Chapter 8, where the automatic link generation problem is formulated as a ranking problem, we use a P-R plot to combine the precision-recall scores. In Chapter 9, where the linking problem is formulated as a classification problem, we use the F-measure to combine the two scores.

2.4.3 Statistical significance testing

While comparing system performance in terms of certain evaluation measures, significance tests are often used to determine whether or not the observed differences in system performance is due to chance.

A significance test consists of the following essential ingredients [24, 230].

1. A test statistic or criterion by which to judge the two systems, e.g., the difference in the mean of an IR metric.

2. A distribution of the test statistic given a *null hypothesis.* A typical null hypothesis is that there is no difference between our two systems.

3. A significance level that is computed by taking the value of the test statistic for our experimental systems and determining how likely a value that is large or lager could have occurred under the null hypothesis. This probability of the experimental criterion score given the distribution created by null hypothesis is known as the *p-value*.

Commonly used significance tests include the paired Student's t-test, the paired Wilcoxon signed rank test [259] and the sign test [116, 230]. These tests differ in their assumptions about the distribution of the data being tested. For example, the t-test requires that the two samples, i.e., the evaluation results of the two systems being compared, follow a normal distribution and have equal variance, while the Wilcoxon signed rank test and the sign test are non-parametric tests and do not require these conditions to be satisfied. Sanderson and Zobel [214] find that the t-test tends to be more reliable than the sign test or Wilcoxon test, even when some of the assumptions are violated. Further, significant results found on 25 or less queries are not guaranteed to be repeatable on other set of queries. Finally, as pointed out by Keen [131], the statistical and

practical significance of the differences should be carefully assessed. Differences that are not statistically significant can still be important if they occur repeatedly in many different contexts [116].

In this thesis, we use the paired t-test for significance testing. Our null hypothesis is: there is no difference between the performance of the two systems being compared, where the performance is evaluated using an evaluation measure discussed in the previous section. We set a critical value of 0.05 over the p-value to determine whether a difference is significant. That is, a p-value smaller than 0.05 indicates a significant difference and a rejection of the null hypothesis.

Part I

Topical Coherence

Chapter 3

A Measure for Topical Coherence

The first research theme we address is *topical coherence*. The topical coherence of a set of documents is associated with the following two properties of the document set: (i) the number of topics covered, and (ii) the degree to which the documents are focused on these topics. Roughly put, a document set that covers a single topic is topically more coherent than a document set that covers multiple topics; and for two document sets both covering multiple topics, the set dedicating the majority of the documents to one topic is more coherent than the set "equally" discussing each of the topics.

While our aim is to quantify these properties, it is clear that both properties are relative concepts. Particularly, they are relative concepts with respect to the granularity of the topics we consider. For example, a set of documents discussing the topic "World of Warcraft" (WoW), an online game, covers various aspects of the topic: gameplay, game development, game community, etc., at a high level of granularity, the document set contains one topic, i.e., WoW, while at a lower granularity level, multiple topics related to the game are discussed. Similarly, the documents can be seen as focused on the general topic of WoW, but less focussed with respect to each of the sub-topics. From the above example we see that quantifying these properties with an absolute value need not be very meaningful as it can change easily when the granularity of topics changes. In order to determine the topical coherence of a document set, we need a point of *reference* in relation to which the level of topical granularity is considered.

In this chapter, we introduce a *coherence score* that captures the topical coherence of a set of documents using a document set randomly drawn from a background collection as reference. We use the vector-space model to represent documents and topics are represented by clusters of documents. We then take a clustering perspective and determine the topical coherence of a set of documents by comparing its clustering structure against that of the reference set. We investigate the properties of the proposed score both theoretically and empirically. Recall the research questions we formulated in Chapter 1, which we aim to answer in this chapter:

RQ1a. How do we measure the topical coherence of a set of documents?

RQ1b. Can the coherence score we propose effectively reflect the topical coherence of a set of documents?

3.1 The coherence score

The coherence score we propose is a measure for the relative tightness of the clustering structure of a specific set of data as compared to the background collection. We derive our inspiration from the *expression coherence* score used in the genetics literature [189].

Given a set of documents $D = \{d_i\}_{i=1}^{M}$, which is drawn from a background collection C, i.e., $D \subseteq C$, we define the coherence score as the proportion of "coherent" pairs of documents with respect to the total number of document pairs within D. The criterion of being a "coherent" pair is that the similarity between the two documents in the pair should meet or exceed a given threshold. Formally, given the document set D and a threshold τ, we have:

$$\delta(d_i,d_j) = \begin{cases} 1 & \text{if } \textit{similarity}(d_i,d_j) \geq \tau, \\ 0 & \text{otherwise.} \end{cases} \quad i \neq j \in \{1,\ldots,M\} \tag{3.1}$$

where the similarity between documents d_i and d_j is a similarity or distance measure[1] describing the semantic closeness of the two documents. It can be any similarity or distance measures, depending on the applications.

The *coherence score* (Co) of the document set D is then defined as

$$Co(D) = \frac{\sum_{i\neq j\in\{1,\ldots,M\}} \delta(d_i,d_j)}{\frac{1}{2}M(M-1)}. \tag{3.2}$$

For D with single document, we define its coherence score as 0.[2]

From the above definition, we can see that the threshold τ is an important parameter to determine the coherence score. As stated previously, the coherence score measures the relative tightness of the clustering structure of a set of documents compared to the background collection; the threshold τ actually establishes the connection between the two.

3.1.1 Design choices

Given the definition of the coherence score, the following free parameters need to be determined in order to calculate the coherence score: the representation of documents, the similarity measure and the parameter κ that is used to determine the threshold τ.

[1] Note that if a distance measure is used, the criterion of a pair of documents being "coherent" is that the distance between the pair should be *smaller than* a given threshold.

[2] Note that Eq.3.2 requires $i \neq j$, which implies that in the case D only contains one document, the coherence score is not properly defined. Although one may argue that one document can be seen as coherent to itself, we prefer to assign a score of 0. From a clustering point of view, it is trivial to have each single document being a cluster.

In order to obtain the value of τ, we randomly sample n documents from the background collection C and calculate the pair-wise similarities. Then, we order the $\frac{1}{2}n(n-1)$ similarity scores and take the value of the score at the top $\kappa\%$ fraction as the value of τ'. That is, we assume that $\kappa\%$ pairs from the set of documents randomly drawn from the background collection are "coherent" pairs. We repeat this sampling for r runs and for different values of n and approximate the actual τ by taking the mean value of τ's from all these different runs. Any pairs of documents whose similarity meets or exceeds τ are considered to be "coherent" pairs. For the value of κ, we heuristically set it to 5.

Throughout this thesis, to calculate coherence scores, we represent documents using the vector-space model with a TF-IDF term weighting scheme and use the cosine similarity as the similarity measure. On the one hand, these choices are made for simplicity. Both the document representation and the cosine similarity are widely used and shown to be effective in various IR applications [105, 271], which makes them a "safe" choice and a good starting point. On the other hand, we have found that throughout our experiments on various tasks, coherence scores calculated with these choices are effective [100, 101, 102, 104].

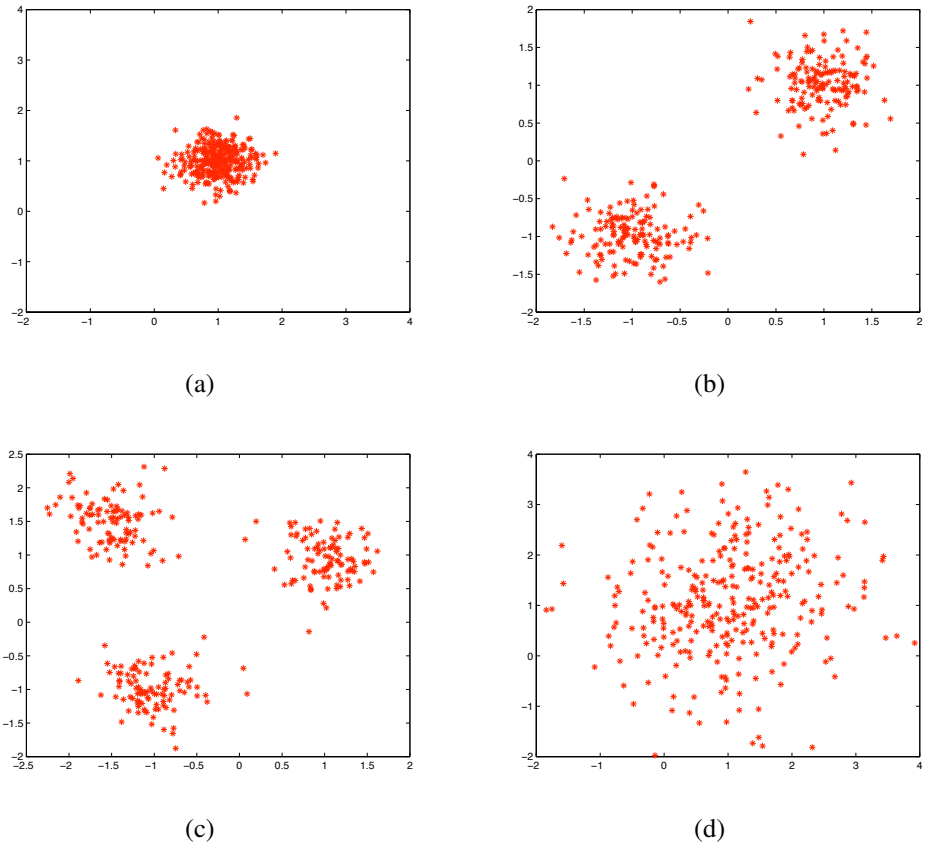
3.1.2 A toy example

The properties of the coherence score, and thereby its capacity to represent clustering structure, can be visualized by making use of a toy example. We generate four artificial data sets: (a), (b), (c) and (d), with different clustering structures. Data set (a) consists of a tight cluster. Data sets (b) and (c) consist of 2 and 3 sub-clusters, respectively. Data set (d) consists of one loose cluster which is generated by a normal distribution with large variance and we consider it to be the background set (or a random set). Figure 3.1 illustrates these four data sets.

The variance is a commonly used measure for the degree to which a data set is "spreadness:" the smaller the variance, the more tightly the data points are gathered. We calculate the coherence score and the total variance for the four data sets (a), (b), (c) and (d). Table 3.1 shows the results. We can see that, ranked in terms of total variance, we have $(a) > (d) > (b) > (c)$; while in terms of coherence, we have $(a) > (b) > (c) > (d)$, whereby ">" means a tighter structure. Thus, the coherence score differs from the variance score in its ability to differentiate between data sets with and without clustering structure. From this toy example we can see that the coherence score prefers a structured data set to a random set, and among structured data sets, it prefers the sets with fewer sub-clusters.

datasets	(a)	(b)	(c)	(d)
coherence score	0.0092	0.0056	0.0034	0.0006
total variance	0.1748	2.1728	2.5315	2.1227

Table 3.1: The coherence score and the total variance of the toy data sets.



(a) (b)

(c) (d)

Figure 3.1: A toy example. (a) One random sample from a normal distribution with $\mu = (1,1)$, $\sigma = 0.3$; (b) two random samples from a normal distribution with $\mu_1 = (-1,1)$, $\mu_2 = (1,-1)$, $\sigma_1 = \sigma_2 = 0.5$; (c) three random samples from a normal distribution with $\mu_1 = (-1,1)$, $\mu_2 = (1,-1)$, $\mu_3 = (-1.5,1.5)$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.5$; (d) one random sample from a normal distribution with $\mu = (1,1)$, $\sigma = 1$.

3.2 Impact of the size of document sets

As the coherence score is defined as the proportion of "coherent" document pairs with respect to the total number of document pairs in the set, one important issue one may be concerned with is the following: do the sizes of document sets have an impact on the coherence score? Further, are the coherence scores comparable across document sets of different sizes?

Below, we define a case with strict assumptions about the clustering structure of the document sets. With this restricted case we aim to provide insight to the impact of the size of document sets on the coherence score.

First we introduce our notation. For a given set of documents D and a given threshold τ, if $\forall(d_i,d_j) \in D$, $similarity(d_i,d_j) \geq \tau$, we call it a *self coherent* set. Further, for two document sets D_1 and D_2, if $\forall(d_i,d_j)$ where $d_i \in D_1$ and $d_j \in D_2$, $similarity(d_i,d_j) <$

τ, we say that these two sets are *mutually exclusive*.

Assume we have two document sets $D_1 = \{d_i\}_{i=1}^{M_1}$ and $D_2 = \{d_j\}_{j=1}^{M_2}$, with $M_1, M_2 > 1$. Each data set (D_1 or D_2) can be divided into two subsets $A = \{d_n\}_{n=1}^{N}$ and $B = \{d_m\}_{m=1}^{M-N}$, where $N \geq M-N$. We call set A the *dominant* subset of D.

Let τ be given and assume the following conditions hold:

Condition 1 $\frac{N_1}{M_1} = \frac{N_2}{M_2}$.

Condition 2 A_1, B_1, A_2, B_2 are self coherent or singleton;

Condition 3 A_1 and B_1 are mutually exclusive; A_2 and B_2 are mutually exclusive;

We establish the following property of the coherence score under the above conditions:
Property 1. $Co(D_1) > Co(D_2)$ if and only if $M_1 > M_2$

Proof. According to Condition 2 and 3, we know that

$$Co(D) = \frac{1}{2}\left(N(N-1) + (M-N)(M-N-1)\right)\frac{2}{M(M-1)} \tag{3.3}$$

$$= 1 + \frac{2M}{M-1}\left(\left(\frac{N}{M}\right)^2 - \frac{N}{M}\right) \tag{3.4}$$

According to Eq. 3.4, we have

$$Co(D_1) - Co(D_2) = \frac{2M_1}{M_1-1}\left(\left(\frac{N_1}{M_1}\right)^2 - \frac{N_1}{M_1}\right) - \frac{2M_2}{M_2-1}\left(\left(\frac{N_2}{M_2}\right)^2 - \frac{N_2}{M_2}\right). \tag{3.5}$$

According to Condition 1,

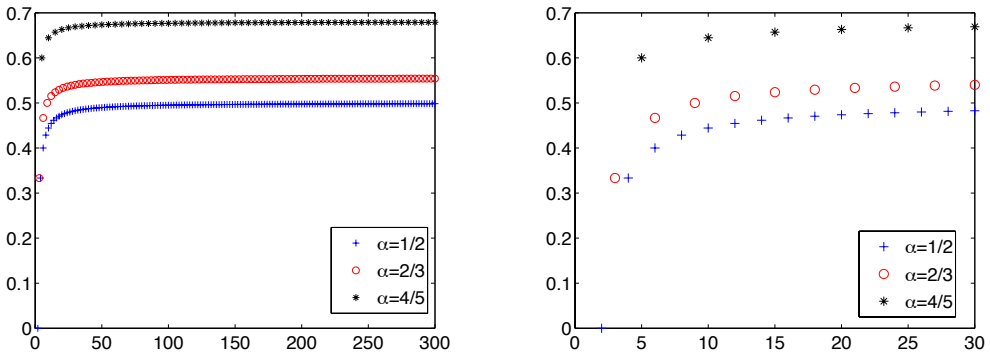
$$\left(\frac{N_1}{M_1}\right)^2 - \frac{N_1}{M_1} = \left(\frac{N_2}{M_2}\right)^2 - \frac{N_2}{M_2} \leq 0.$$

Let $y = \left(\frac{N_1}{M_1}\right)^2 - \frac{N_1}{M_1} = \left(\frac{N_2}{M_2}\right)^2 - \frac{N_2}{M_2}$. Then Eq. 3.5 can be reduced to

$$Co(D_1) - Co(D_2) = y \cdot \frac{M_2 - M_1}{(M_1-1)(M_2-1)}. \tag{3.6}$$

Since $y \leq 0$ and $M_1, M_2 > 1$, if $M_1 > M_2$, we have $Co(D_1) > Co(D_2)$, and if $Co(D_1) > Co(D_2)$, we have $M_1 > M_2$. □

The above property of the coherence score implies that the size of a document set has an impact on the coherence score of the document set. This property is intuitively appealing: when 12 out of 20 documents focus on a single topic, it is perceived as more strongly "topically focused" than in a case where only 3 out of 5 documents focus on the topic. In practice, we find that as the size of the document sets increase, their impact



(a) (b)

Figure 3.2: Impact of the size of document sets on the coherence score. The X-axis is the number of documents in a document set, and the Y-axis is the corresponding coherence score. Figure 3.2(a) shows the coherence scores over range [0, 300] and in Figure 3.2(b) we zoom in on the first part of Figure 3.2(a) over range [0, 30].

on the coherence score diminishes. In Figure 3.2, we show the impact of the size of document sets on the coherence score. Assume we have document sets with varying sizes $M = 2, \ldots, N$, and each document set contains two self-coherent and mutually exclusive subsets A of size αN and subsets B of size $(1-\alpha)N$, where α indicates the proportion of documents in A. In Figure 3.2 we show the change of coherence score with respect to the change of the size of the data sets. We show the cases when α is set to 1/2, 2/3, and 4/5. We see that the coherence scores converges quickly as the size of the document sets increases. When we zoom in, we see that in Figure 3.2(b), when the size of document sets reaches 15, the coherence score stabilizes.

3.3 Experimental evaluation of the coherence score

3.3.1 Experimental setup

In order to test the power and reliability of the coherence score in measuring the topical structure of textual data, we perform experiments using simulated data, which is constructed using real world data whose clustering structure is kept strictly under control.

Test collections

We first detail the collections we use in this experiment. We list the TREC test collections we use in Table 3.2 alongside their main document types and the topics that are used in the experiments. Our aim in selecting these particular collections is the different types of documents they contain: the blog collection contains individual blog posts, that is, user generated content. The other two collections contain formal, edited

collection	document type	queries
AP89+88	news	1–200
Robust04	news, governmental	301–450, 601–700
TRECBlog06	blog posts	851–950

Table 3.2: Collection characteristics and used queries.

content from two sources: news and governmental pages. User generated content, with its lack of editors and top-down rules, differs from more formal, edited content in various ways, for example (i) spelling and grammatical errors are more common in blogs because of the lack of editors, (ii) the language usage in blogs is more diverse, whereas formal content often uses a fairly narrow vocabulary. These collections allow us to check whether we can use one measure of topical consistency for different types of documents.

Simulating document sets

We simulate four data sets each containing 60 documents taken from the TREC test collections. The first three data sets are generated by randomly sampling 1, 2, and 3 queries from the TREC queries, and extracting the relevant documents from the TREC qrels. In this way, we control the topical structure of the document set by varying the number of topics it contains. The fourth data set is a random set, sampled directly from the background collection. We calculate the coherence score for each data set. The construction procedure for the four data sets is repeated 100 times.

3.3.2 Results

Table 3.3 shows the average coherence scores for 100 runs on different TREC collections. The results in Table 3.3 reveal that on average, data sets with a 1-topic cluster obviously have higher coherence scores than data sets with 2- and 3- topic clusters and the random data set. Although the collections are composed of documents of a different nature, i.e., news articles as well as user generated content (blogs), the behavior of the coherence score is consistent across data sets. This experiment shows that the coherence score does indeed reflect the topical structure of a set of documents.

3.4 Discussion and conclusion

In this chapter, we introduced a coherence score that measures the topical coherence present among a set of documents by evaluating the relative tightness of the clustering structure of the document set as compared to a background collection.

We have provided a theoretical analysis of the impact of the size of document sets on the coherence score, which not only gives insights in the behavior of the coherence

	scores	1-topic clusters	2-topic clusters	3-topic clusters	random sets
AP89+88	mean	0.7205	0.4773	0.3900	0.0557
	(var)	(0.0351)	(0.0199)	(0.0284)	(0.0002)
Robust04	mean	0.6947	0.4490	0.3365	0.0457
	(var)	(0.0463)	(0.0114)	(0.0064)	(0.0002)
Blog06	mean	0.6663	0.5215	0.4405	0.0495
	(var)	(0.0378)	(0.0226)	(0.0126)	(0.0003)

Table 3.3: The mean value and variance of the coherence scores for clusters containing different numbers of topics, each setting is sampled 100 times with a cluster size of 60 documents.

score, but also provides implications that need to be taken into account when applied to certain tasks in practice. On top of that, we have empirically evaluated the effectiveness of the coherence score in capturing topical coherence using simulated data. Empirical results suggest that the coherence score is effective in capturing topical structure present in a set of documents.

It is worth mentioning that the coherence score shares certain similarities with the random graph-based cluster tendency tests proposed in the literature [153, 154]. The cluster tendency tests are a type of cluster validating technique that was aimed at determining whether the clusters in a dataset are significantly different from random, since it is certainly inappropriate to impose a clustering structure on a dataset known to be random [59]. The random graph-based cluster tendency tests work as follows. Given a dataset with n data points, each data point corresponds to a vertex of a graph, and clustering corresponds to the procedure of entering edges between pairs of data points, for example, between those whose similarity is higher than a threshold t. The resulting clustering structure of the data points at a given t corresponds to a graph with n vertices and v edges. Then the randomness of the observed structure is captured by comparing it with the probable structures of all possible graphs with n vertices and v edges. Under the random graph hypothesis, each of those graphs is equally probable. A number of quantities can be used to assess the randomness of the resulting structure. For example, the minimum number of edges V that a random graph of n vertices needs in order to become connected. Ling and Killough [154] provided an exact method to calculate tables for the probability of observing a specific value of V under the random graph hypothesis for given n. Often an arbitrary threshold is set to the probabilities to determine whether an observed structure is random, say 0.99 [59].

While the goal of measuring topical coherence and testing cluster tendency is different, the two approaches share the same idea of comparing the clustering structure of a dataset to randomness. The coherence score can be seen as a simplified case where we assume that 5% (given that κ is set to 5%) of the document pairs in a random set should be connected. Using the similarity threshold obtained under this assumption, for a given document set, we observe the percentage of connected document pairs in the

data set. The higher the percentage of observed connected document pairs, the more coherent the document set is. Instead of using the random graphs, we use the background collection as an approximation of randomness. While theoretically appealing, the idea of using random graphs with equal probability is questionable, as it assumes that any random document pair has the same probability to be connected, that is, has the same probability of having a similarity higher than a threshold, which may only occur if all document pairs have the same degree of similarity. Nevertheless, it would be interesting future work to consider using other types of random graph models, i.e., different probability distributions on graphs, as a reference of randomness.

In the next two chapters, we will use the coherence score in the setting of two IR tasks, where topical coherence plays an important role.

Chapter 4

Blog Retrieval: Topical Consistency among Documents

The task on which we focus in this chapter is *blog feed retrieval*, also called blog distillation [162]. The blog feed retrieval task is defined as *identifying blogs that show a central, recurring interest in a given topic*. The task has two main characteristics: first, the retrieval units are blogs rather than single posts; second, in order to be considered as relevant, a blog should not just mention the topic of the user query sporadically, but rather it must contain a significant number of posts concerning this topic. An effective approach to blog feed retrieval should take both of these characteristics into account.

Within this context, we investigate whether our coherence score can be exploited to model topical consistency of a blog and thereby improve retrieval effectiveness in the blog feed retrieval task. We start with a brief introduction of the blog feed retrieval task and recall the research questions we have outlined in Chapter 1.

4.1 Introduction

The amount of user generated content available on-line is already voluminous, and it continues to grow on a daily basis. User generated content is not regulated by top-down rules, leaving users free to decide (i) what to write about (topics), (ii) how to write (writing style, language), and (iii) when to write (time of day, regularity). Since user generated content is produced without editorial supervision, standards and conventions that otherwise dictate the form and consistency of written prose, cannot be assumed to be upheld. A specific type of user generated content, blogs (syndicated web journals), has shown a particularly spectacular rise. Currently, bloggers worldwide generate content at a rate in the order of one million new posts per day.[1] With this ever increasing amount of information available in the blogosphere, the need for intelligent access facilities is clear. The information needs of users searching the blogosphere fall into two general categories: the need to find individual blog posts regarding a topic, or the need

[1] `http://technorati.com/state-of-the-blogosphere/`

to identify blogs that frequently publish posts on a given topic. These categories mirror the short term vs. long term interest distinction observed by Mishne and de Rijke [180] in their study of blog search behavior. Although currently most focus is on finding blog posts, some systems offer the possibility to search for full blogs, alongside post-level retrieval functionality [72]. Searchers can use blog search to identify blog feeds they would like to add to their feed readers.

Two key features set blog content apart from conventional web content and necessitate that dedicated retrieval algorithms and approaches be developed for blogs. The first is the strong social aspect of blog content, most readily noticeable in the use of blog rolls, user assigned tags and, especially, comments to posts. The second, and the one most relevant to the current context, is the noisiness of the data in the blogosphere. We identify two levels at which blog content is noisy: (i) the blog post level and (ii) the blog level. At the post level, noise expresses itself in unexpected language usage, spelling and grammatical errors, non-language characters (e.g., emoticons), and mixed data types (pictures, video, text). At the blog level, the noise can be characterized as *topical noise*, which tends to be semantic rather than lexical or structural. A blog can (and most likely will) be about different topics. As an illustration of different levels of topical noise in blogs, consider Figures 4.1(a) and 4.1(b), where two blogs treating the subject of vegetable gardening are displayed in the NetVibes[2] feedreader. In the blog in Figure 4.1(b), the blogger digresses from the topic of vegetable gardening to write about other topics. Dealing effectively with this type of topical noise is critical for improving performance on blog feed retrieval, since blogs with topical noise show less consistent interest in particular subjects and are therefore a priori less likely to be appreciated by users in the setting of the blog feed retrieval task.

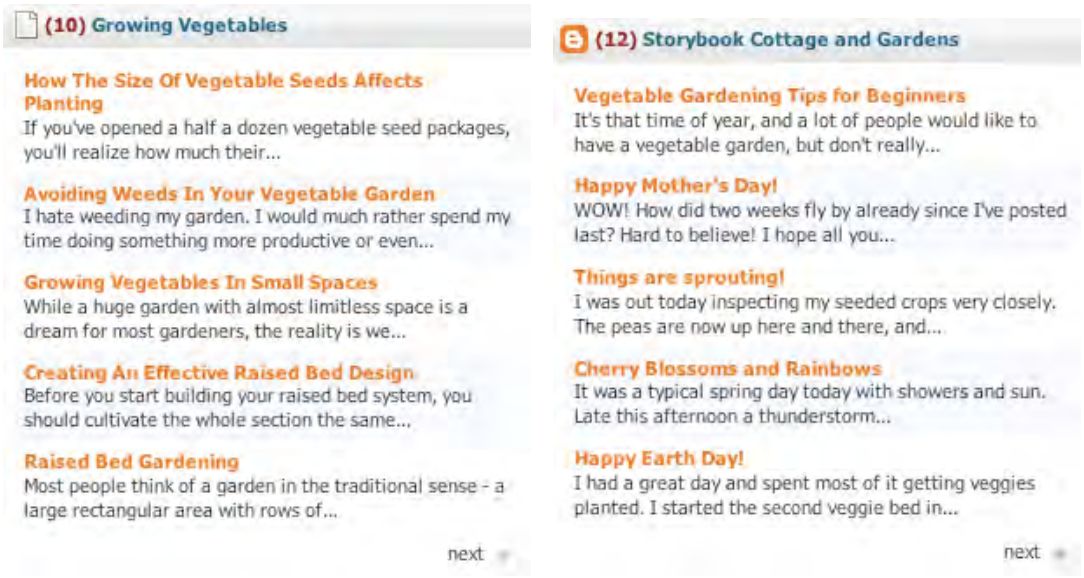
How can we measure topical noise? Specifically, how can we measure it in blogs? The characteristics of the blog feed retrieval task combined with the challenge presented by noisy data require an approach that is both flexible and sufficiently robust. We view blog feed retrieval as an association finding task: which blogger is most closely associated with the given topic? And: how consistent is this blogger regarding the topic? To address the first issue, we adopt the language modeling approach used in expert retrieval [13, 256]. To tackle the second issue—the core issue addressed in this chapter—, we integrate the *coherence score* into this language modeling-based approach. The coherence score measures the topical clustering structure of a blog. Loose clustering reflects topical diffuseness and signals the presence of topical noise in the blog as defined in Chapter 3. In contrast, tight clustering indicates that the blog remains focused on one or a few central themes.

Given these issues, we explore the following three dimensions in this chapter, which we formulate as research questions:

RQ2a. How do we measure topical consistency for a blog?

RQ2b. How can we use the coherence score in our blog retrieval process?

[2] `http://www.netvibes.com`



(a) (b)

Figure 4.1: (a) Example of a blog with little to no topical noise. (b) Example of a blog with a moderate level of topical noise.

RQ2c Given that the collection we use in our experiments only provides us with a sample of blog posts generated by the underlying blog models, how does the sample size influence the estimation of the coherence and how does this influence blog feed retrieval?

For RQ2a, we offer our coherence score as a solution; we compare it against lexical cohesion, a standard measure for determining the diversity of topics discussed in a text. For RQ2b, we compare a number of options, ranging from treating the coherence score as a simple prior to modeling it as a multiplicative factor whose contribution is a function of the retrieval status value of a blog. The final question RQ2c is addressed using an experimental exploration.

Our main finding is that our proposed coherence score can estimate the topical noise present in blogs. Moreover, it can help combat the topical noise present in blogs when it is weighted with the initial retrieval score, preventing blogs that display tight topic structure, but that are not relevant to the query, from rising to the top of the result list. In addition, we find that a minimum of 20 posts is required to get a proper estimate of the coherence of a blog, regardless of the actual size of the blog. This finding is supported by blog feed retrieval results: The coherence score reaches its optimal performance increase when a substantial number of posts (> 20) have been written in a blog.

The remainder of the chapter is organized as follows: In Section 4.2 we study our proposed coherence score as compared against the well-known lexical cohesion measure. In Section 4.3 we detail the modeling of blog feed retrieval and the integration

of coherence in this framework. Section 4.4 specifies our experimental settings, and we discuss the results of the experiments in Section 4.5. In Section 4.6 we analyze our experimental findings, before concluding in Section 4.7.

4.2 Topical consistency measures

This section discusses two methods of capturing the topical consistency of a text. In Chapter 3 we have already introduced our coherence score as a measure of topical coherence. Here, we introduce *lexical cohesion*, a familiar text analysis approach that uses information about the semantic relatedness of words to capture the topical structure of a text. We then compare our coherence score against lexical cohesion. Evidence emerges that the advantages of lexical cohesion are outweighed by its shortcomings. In particular we comment on its lack of sensitivity to topical hierarchy.

4.2.1 Lexical cohesion

The concept of cohesion [82] is used in text analysis to describe the topical relationships between various units of text. Cohesion is a set of characteristics that conspire to make text "stick together" topically [15]. Lexical cohesion measures cohesion by examining the semantic relationships between the content words used in a text [182]. Lexical cohesion is easy to identify [15] and can be calculated automatically using an appropriate linguistic resource such as a thesaurus. Semantically similar words (usually nouns) occurring in close proximity to one another build lexical chains, which indicate that a unit of text is about the same topic [182]. Lexical chains form the basis for models of lexical cohesion [15, 182, 236]. A primitive form of lexical cohesion does not make use of similar lexical words, but rather measures repetition of the same word form or forms. The *cohesiveness filter* proposed by Amitay et al. [8] encodes an entropy-based measure of query-word repetition patterns and, is an example of this primitive form of lexical cohesion. The disappointing results of this filter as applied to the task of identifying topically focused web pages in the TREC-2003 Web Track topic distillation task motivate us to turn our consideration to full-fledged forms of lexical cohesion that look beyond word-form repetition and make use of external resources to derive information concerning lexical similarity.

A priori, lexical cohesion is an appealing approach to capturing topical consistency. It is intuitive that the topical diversity of a text is reflected in the number of distinct topics it discusses. The number of topics in a text, in turn, is reflected by the number of lexical chains of words with similar meanings that the text contains. The low-noise blog excerpt in Figure. 4.1(a) and the moderate-noise blog excerpt in Figure. 4.1(b) are convenient examples that provide an impression of how lexical chains capture topical consistency. We generate *strong lexical chains* from these blog excerpts using the LexicalChain application of the Electronic Lexical Knowledge Base (ELKB),[3] which

[3] `http://www.nzdl.org/ELKB`

is based on Roget's Thesaurus and implements the algorithm proposed by Barzilay and Elhadad [15]. The chains are shown in Table 4.1. The LexicalChain algorithm computes lexical chains by clustering words that are both semantically similar and near to each other in the text. *Chain score* is the length of the chain as measured by the number of words it contains weighted with a factor reflecting the number of repeated words. *Strong chains* are defined as chains that have a score greater than the mean score plus two standard deviations. The highest frequency member of a chain is defined to be its *keyword*. From Table 4.1, it can be observed that the low-noise blog excerpt generates eight strong lexical chains, seven of which have a unique keyword. The moderate-noise blog, on the other hand, generates nine strong lexical chains. The difference in chain number reflects human intuitions about the topical diversity of the two blogs. The difference is not strikingly large, but still serves to illustrate the way in which intuitions of topical diversity are related to the number of topics as reflected by the number of lexical chains a text contains. Other five post excerpts of the same blogs display similar differences in chain number.

In addition to providing an impression of how lexical cohesion works, this example also illustrates one of its shortcomings. Lexical cohesion is sensitive to the progression of topics in a text, but is rather blind to their hierarchical structure. Where humans may differentiate between a central and a subordinate topic, the LexicalChain algorithm produces two lexical chains of approximately the same length. For example, in Table 4.1 it can be seen that in the low-noise topical blog, a chain with the keyword "soil" is produced, which is a plausible central topic of the blog. A chain with the keyword "space" is also produced, which arises due to mention of spatial concepts in various contexts, but is less likely to be understood as an actual topic of the blog. It is challenging to determine the topical consistency of a text collection by using lexical chains to count the number of distinct topics occurring, since it is not readily obvious which chains to count as representing central topics of the text.

The problem of distinguishing central from subordinate topics can be circumvented by setting aside the chain-based lexical cohesion approach, and instead looking directly at the inherent clustering structure of the collection, i.e., the topic groups that emerge when the documents in the collection are compared to each other. That said, the coherence score we proposed in Chapter 3 is such a measure that captures the clustering structure of the collection. In the next section, we revisit the coherence score and compare it to lexical cohesion.

4.2.2 Coherence score versus lexical cohesion

In Section 3.3 on page 40 we have seen that the coherence score is able to capture the clustering structure of data, and in particular, the topical consistency of text. The coherence score holds clear potential for capturing the topical consistency of user generated content. Here, in order to get a direct impression of the effectiveness of the coherence score as compared to the lexical cohesion, we calculate the coherence score for the blog examples discussed above. The two blogs generate coherence scores consistent

Strong lexical chains the low-noise blog excerpt in Figure. 4.1(a) (five posts of "Growing Vegetables")
garden (plant, sow, bed, seed, gardeners, weed, plants, planted, weeding, landscape, beds, sown, cultivate, seeds, weeds, garden)
soil (building, yard, side, ground, stone, walk, rows, soil)
raised (realize, fruit, raised, harvesting, clearing, finding, crops, bring, light, crop, produce)
space (reach, keeping, wide, spacing, foreign, space, spread)
easy (leg, sitting, easy, easier, maintain, summer, arm, proper, giving, maintaining)
grow (time, grow, half, growing)
deep (huge, sizes, deal, larger, run, deep, size)
grow (discourage, start, grow, care, growing)
Strong lexical chains in the moderate-noise blog excerpt in Figure. 4.1(b) (five posts of "Storybook Cottage and Gardens")
planted (plant, manure, seed, bed, green, plants, planted, nursery, plot, winter, beds, seeded, hoping, gardening, dig, seeds, garden)
sprouts (fill, biggest, wide, growing, blew, putting, sprouting, spring, pot, grown, full, grow, sprouts, sprout, develop, pots)
bit (dish, root, breakfast, crop, dinner, cookie, super, picking, bit, takes, beets, foods, food, lettuce, crops, eating, eat, pick, square)
row (fit, warm, thunderstorm, ran, fly, heads, weather, sets, heading, row, rows)
left (post, yellow, double, posted, reference, spot, typical, figure, red, forward, blue, left, notes, note)
time (time, woke, days, beans, day, fun, tapes, Day)
starts (starts, start, die, starting, started)
batch (showers, lots, batch, lot, pack, packs, closely)
plans (plans, plan, wire, advise, suggest, plain, explains, planned, advice)

Table 4.1: The low-noise blog excerpt (cf. Figure. 4.1(a)) generates seven unique strong lexical chains, while the moderate-noise excerpt (cf. Figure. 4.1(b)) generates nine. Chains are ordered by decreasing chain strength; keywords are shown in bold.

with our expectations. The coherence score of the blog excerpt with low topical noise (cf. Figure. 4.1(a)) is 0.5 compared to 0.3 for the blog excerpt with moderate topical noise (cf. Figure. 4.1(b)).

What are the advantages of using coherence score as a measure of topical consistency for blogs compared to lexical cohesion? First, the coherence score relies only on the statistics derived from the collection and is independent of any external resources. In order to calculate alternate measures such as lexical cohesion, an external knowledge resource such as a thesaurus or lexical database such as WordNet [66] is necessary. Dependence on external resources raises several issues. External resources often fail to be up-to-date with regard to proper nouns [236], which is especially needed in a fast-changing environment like the blogosphere. Further, we must be able to filter our collection and regard blogs written only in languages covered by available resources, a challenging task in face of the fact that some bloggers switch languages while posting. In these respects, using the coherence score offers clear benefits of independence and flexibility.

Second, the coherence score does not require optimization of parameter settings. For the coherence score, the only parameter is the threshold τ, which defines the "non-randomness" for a given collection. Recall that τ is set by sampling the background collection for a given κ. Although κ is determined heuristically, our previous experiments show that the value 0.05 is quite stable. Coherence is thus easier to apply than measures such as lexical cohesion. In order to build lexical chains, the setting of two parameters is required: a threshold on the semantic relatedness of two words and a threshold on the physical distance, i.e., the number of words separating them in the running text. These parameters determine whether a word should be added to an existing chain or start a new chain [236]. Presumably, parameter settings would have to be

Total number of blogs:	83,320				
blog length	< 10	10–50	50–100	100–500	> 500
number of blogs	21,290	42,085	15,338	4,514	103

Table 4.2: Distribution of the blog lengths, i.e., number of posts contained in a blog.

re-optimized for a new corpus.

Third, the coherence score directly captures the clustering structure of the collection. For this reason, it is not necessary to be concerned about identifying individual topics or their relative importance in the blog. As discussed above, a lexical cohesion measure based on lexical chains encounters the challenge of distinguishing chains representing central topics from chains representing subordinate topics. Although we do not exclude the possibility that further development work would allow this issue to be addressed, the coherence score approach offers the advantage of circumventing the issue entirely.

Fourth, the coherence is more efficient to compute. Its computational complexity is $O(s \cdot n^2)$, while the complexity of a typical lexical chain algorithm is $O(s^2 \cdot n^2)$, where s is the average length of the individual documents in words and n is the number of documents in the document set on which the coherence measure is performed. Although in practice the computational complexity of the calculation of lexical chains can be kept well below its theoretical limit, it still fails to be competitive with that of the coherence score. For our experiments, we calculate the coherence score for the set of blog posts in a given blog. The coherence score is calculated offline at indexing time, i.e., we calculate the scores once for all blogs in the collection. With our implementation, the calculation of pairwise cosine similarity scores takes around 1.5 seconds for 500 documents. Table 4.2 shows the distribution of the number of posts in blogs, which provides an impression of the feasibility of our approach.

Finally, given the definition of the coherence score in Chapter 3, the application of coherence score is not even limited to text data. Information other than words such as the structure of the documents, hyperlinks contained in the webpages, etc., could be easily integrated.

These advantages provide motivation for us to leave aside consideration of measures with the disadvantages of lexical cohesion and continue our investigation by testing the efficacy of the coherence score. In particular, we investigate whether the coherence score can be exploited to model topic consistency and improve retrieval in the blog feed retrieval task.

4.3 Using coherence in the setting of blog feed retrieval

In this section we detail the modeling of the task we address: modeling topical noise in user generated content. To this end, we first explain our blog feed retrieval modeling framework in Section 4.3.1; after that we introduce alternative ways of incorporating

the coherence score in this framework (Section 4.3.2).

4.3.1 Blog retrieval model

Our approach to modeling blog feed retrieval, first introduced in [13], is based on expert retrieval models [12]. As indexing unit we use individual blog posts. We have three reasons for this: (i) to allow for easy incremental indexing, (ii) for presentation of retrieval results posts are natural units, and (iii) the most important reason, to allow the use of one index for both blog post and blog feed search [256].

We adopt a probabilistic approach to the task of determining relevance of blogs to the user query and formulate the task as follows: *what is the probability of a blog being relevant given the query topic q?* In other words, we estimate $p(blog|q)$, and rank blogs according to this probability. Since a query generally consists of only a few terms, often under-representing the information need that gave rise to it, Bayes' Theorem is applied in order to achieve a more accurate estimate:

$$p(blog|q) = \frac{p(q|blog) \cdot p(blog)}{p(q)}, \tag{4.1}$$

where $p(blog)$ is the probability of a blog: in our baseline approach $p(blog)$ is assumed to be uniform, that is $p(blog) = |blog|^{-1}$, where $|blog|$ is the number of blogs in the collection; other ways of estimating $p(blog)$ are detailed in Section 4.3.2. The component $p(q)$ indicates the probability of a query. In the remainder of the chapter, we refer to the Retrieval Status Value (*RSV*) rather than to $p(blog|q)$. This terminological shift is necessary since our experiments involve incorporating scores into $p(blog|q)$ that have the same scale as probabilities, but are not otherwise true probabilities.

Following a common practice in language modeling approaches, $p(q)$ is discarded as it does not affect the ranking of the results (for a given query q). However, when the impact of the coherence score is taken to be a function of the *RSV* (as we will discuss in Section 4.3.2), the normalization term is necessary in order to ensure that the weight of the coherence score is compatible across queries. A non-normalized *RSV* will impose an unwanted limitation of the domain and thereby also the range of the coherence score function.

In our experiments, we apply the full Bayes' Theorem, which leads to the estimation of the probability $p(q)$. To estimate $p(q)$ we adopt the method used by Lavrenko and Croft [147], who estimate the probability of a term $p(w)$ is with the following equation:

$$p(w) = \sum_{m \in M} p(w|m)p(m), \tag{4.2}$$

where w is a term and M is a set of language models derived from top ranked documents. We can translate this equation to our blog feed retrieval model by replacing $p(w)$ with $p(q)$ and M with B, a set of blogs. We end up with Eq. 4.3:

$$p(q) = \sum_{blog \in B} p(q|blog)p(blog). \tag{4.3}$$

We set B to be the top 200 results, i.e., retrieved blogs, for query q so as to estimate $p(q)$.

Next, we focus on the estimation of the query likelihood, $p(q|blog)$: the likelihood of the topic expressed by the query q given a blog. Query likelihood estimation is accomplished using standard language modeling techniques. We build a textual representation of a blog based on posts that belong to the blog. From this representation we estimate the probability of the query topic given the blog's model. The language modeling framework makes it possible to use blog posts to build associations between queries and blogs in a transparent and principled manner.

Our model represents a blog using a multinomial probability distribution over a vocabulary of terms. For each blog, a blog model θ_{blog} is inferred, such that the probability of a term t given the blog model is $p(t|\theta_{blog})$. The model is then used to predict the likelihood that a blog gives rise to a particular query q. We make the assumption that each query term can be assumed to be sampled identically and independently from the blog model. Applying this assumption, the query likelihood is obtained by multiplying the likelihoods of the individual terms contained in the query:

$$p(q|\theta_{blog}) = \prod_{t \in q} p(t|\theta_{blog})^{n(t,q)}, \tag{4.4}$$

where $n(t,q)$ is the number of times term t is present in query q. In order to prevent data sparseness from resulting in zero query likelihoods, we follow standard procedure and smooth the query likelihood model. The maximum likelihood estimate of the probability of a term given a blog $p(t|blog)$, which is then smoothed with term probabilities $p(t)$ estimated using the background collection:

$$p(t|\theta_{blog}) = \lambda_{blog} \cdot p(t|blog) + (1 - \lambda_{blog}) \cdot p(t). \tag{4.5}$$

In Eq. 4.5, $p(t)$ is the probability of a term in the document repository. The effect of smoothing is to add probability mass to the blog model in proportion to how likely that blog is to be generated (i.e., published) by a generic blogger. We discuss the estimation of the smoothing parameter λ_{blog} in Section 4.4.

The individual blog posts act as a bridge to connect t and the blog, resulting in the following estimate of $p(t|blog)$:

$$p(t|blog) = \sum_{post \in blog} p(t|post, blog) \cdot p(post|blog), \tag{4.6}$$

We make the assumption that the post and the blog are conditionally independent, setting $p(t|post, blog) = p(t|post)$. The importance of a given post within the blog is expressed by $p(post|blog)$. A simple approach to estimating this value is to assume a uniform distribution, i.e., all posts of a blog are weighted equally in terms of importance. Under this assumption, $p(post|blog) = posts(blog)^{-1}$, where $posts(blog)$ is the number of posts in the blog.

4.3.2 Incorporating the coherence score into blog retrieval

Now that we have outlined our blog retrieval framework, we shift our attention to the incorporation of the coherence score in this framework. Before we jump to actually modeling this, we take a step back and look at the relation between the coherence of a blog and its relevance regarding a topic. In case of a (topically) relevant blog, this blog should not be highly favored in the final ranking unless it is *also* topically coherent. On the other hand, if we have a blog that has high topical coherence because it consistently treats a different topic than the relevant topic, we do not want this blog to enjoy an unjustified promotion within the final ranking. Instead, we would like to target a more desirable behavior: blogs that are ranked high for a given topic should enjoy a boost from the coherent score that allows them to maintain their prominence while bottom ranked blogs should be prevented from deriving benefit from their coherence score; in the latter case the chance is greater that they are coherent with respect to non-relevant topics. Finally, documents in between should be given a moderate advantage if their coherence scores are high. We can look at this desirable behavior as *local* re-ranking in contrast to *global* re-ranking, which allows for a document to take a brutal jump from the very bottom to the very top of the final ranking.

A transparent, straightforward integration of coherence in our retrieval framework can be implemented by taking the coherence score of a blog to supply information about query-independent blog relevance, encoded by the blog prior $p(blog)$. As detailed in Section 3.1 on page 36, the coherence score is already a proportion, which means that it is scaled like a probability, and for this reason we can simply estimate

$$p(blog) = Co(blog) \tag{4.7}$$

where $Co(blog)$ is calculated using Eq. 3.2 on page 36, and the threshold τ is estimated to be 0.1, given that κ is set to 0.05 heuristically. In cases where the coherence score of a blog is zero, or when no coherence can be calculated (in the case of one-post blogs), we assign a low probability (0.01). On the one hand we do not want zero probabilities, but on the other we believe these blogs should not receive a high prior probability, since they do not show the recurring interest in a topic.

Although the implementation of coherence as a prior is straightforward, it does not fulfill the properties we discussed in the first paragraph of this section: topically more relevant blogs should receive a solid boost if coherent, less relevant documents should not be affected. In fact, this boils down to weighting the coherence score by some notion of topical relevance. One issue here is that we do not have relevance judgements for our ranked documents. Instead, we use the baseline retrieval score *RSV* of a blog with a uniform prior (viz. Eq. 4.1), as a substitute for judged relevance. We prefer the retrieval score of the blog over an obvious alternative, using the rank of the blog in the retrieval result list. If the rank were used, a small difference in *RSV* could have a disproportionately large impact on the rank, making the weights over-sensitive and unreliable.

In order to capture the desideratum that more relevant blogs receive a bigger boost

from the coherence score, the weights are functions of *RSV*, the baseline retrieval score, and are designed to be monotonically increasing. In particular, we want blogs with *RSV*s close to 0 to receive nearly no contribution from the coherence score while blogs with the highest *RSV*s should receive the full impact from the coherence score. The following functions modify the relation between the coherence weight ($W(\cdot)$) and the *RSV* in a manner consistent with these requirements. We have selected these functions to represent the range of possible relations between *RSV* and coherence score that we believe could potentially be useful.

Linear function (*lin*)

$$W(RSV) = RSV \tag{4.8}$$

Normal distribution (*norm*) with $\mu = 1$ and σ as a free parameter:

$$W(RSV) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(RSV-\mu)^2}{2\sigma^2}\right) \tag{4.9}$$

Quadratic function 1 (*quad1*):

$$W(RSV) = RSV^2 \tag{4.10}$$

Quadratic function 2 (*quad2*):

$$W(RSV) = -(RSV-1)^2 + 1 \tag{4.11}$$

Mixed function of 4.10 and 4.11 (*qmix*) with α as the free parameter:

$$W(RSV) = \begin{cases} RSV^2 & \text{if } RSV < \gamma, \\ -(RSV-1)^2 + 1 & \text{otherwise.} \end{cases} \tag{4.12}$$

This choice of functions allows us to explore a linear relation (Eq. 4.8), a non-linear relation with different rates of increase (Eq. 4.9, 4.10, 4.11), and a combination of different rates of increase (Eq. 4.12). Figure 4.2 shows the curves of these functions in order to provide an intuition of the properties of the functions.

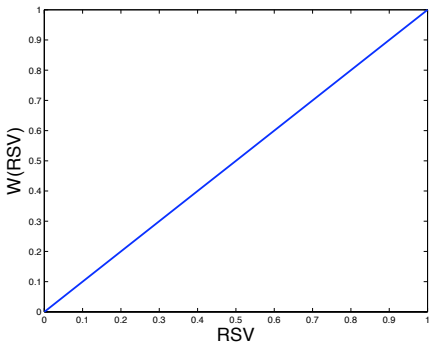
Finally, the weighted coherence score of a blog for a given query is defined as:

$$wCo(blog, query) = W(RSV) \cdot Co(blog) \tag{4.13}$$

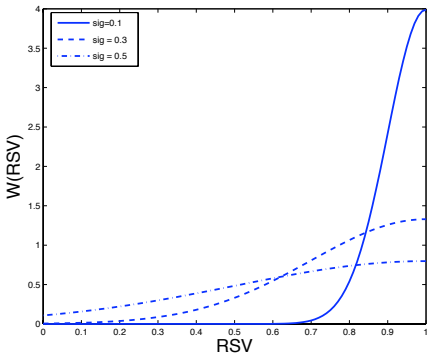
The experimental models use *wCo* as the blog "prior." Substituting it for $p(blog)$ in Eq. 4.1 leaves us with the final ranking equation

$$RSV = \frac{p(q|blog) \cdot wCo(blog, q)}{p(q)}. \tag{4.14}$$

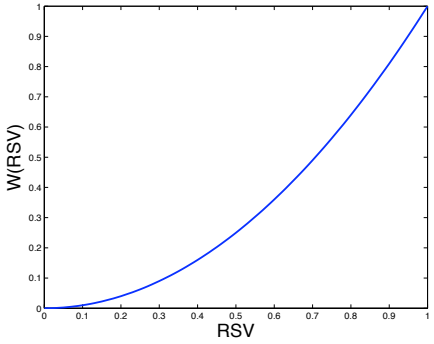
In summary, from our observations on the relation between coherence and relevance, we introduce two main methods for incorporating the coherence score into our retrieval framework: (i) a query-independent method, using $Co(blog)$ directly as $p(blog)$, and (ii) a relevance-dependent method, where $Co(blog)$ is weighted using a function of the *RSV*. The latter method is translated into five weighting functions.



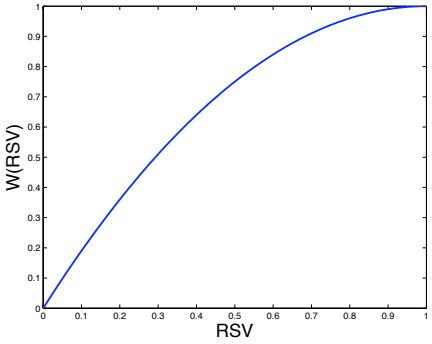
(a) Linear (*lin*)



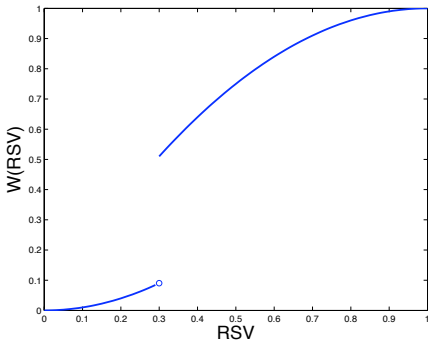
(b) Normal distribution (*norm*)



(c) Quadratic 1 (*quad1*)



(d) Quadratic 2 (*quad2*)



(e) Quadratic mixed (*qmix*) example: $\gamma = 0.3$

Figure 4.2: Weighting functions.

4.4 Experimental setup

Our next aim is to compare the effectiveness of the blog retrieval methods just described. In particular, we aim to answer the following research questions as introduced in Section 4.1:

RQ2a. How do we measure topical consistency for a blog?

RQ2b. How can we use the coherence score in our blog retrieval process?

RQ2c Given that the collection we use in our experiments only provides us with a sample of blog posts generated by the underlying blog models, how does the sample size influence the estimation of the coherence and how does this influence blog feed retrieval?

Before answering these research questions, we detail our experimental setup.

4.4.1 Collection

For our experiments on blog feed retrieval we use the TRECBlog06 collection [161]. The TRECBlog06 corpus was collected by monitoring feeds (blogs) for a period of 11 weeks and downloading html documents behind all permalinks. For each permalink (or blog post or document) the blog ID is registered. For our experiments we did not make use of the syndication information (i.e. RSS feeds). The collection contains 3.2 million blog posts gathered from 100K blogs.

The TREC 2007 Blog track supplies 45 blog feed retrieval topics, also referred to here as queries, and assessments concerning which blogs are relevant to which topics [162]. Topic development and assessment annotation were carried out by the participants of the track. In order to determine the relevance of a blog to a topic, assessors were asked to confirm that a substantial number of blog posts did indeed deal with that topic. For all our runs we make use of the topic field (T) of the topics and discard the longer formulations of the topics (i.e., those contained in the description (D) and narrative (N) fields).

4.4.2 Evaluation metrics and significance

In order to measure the performance of our approach to modeling topical noise in blog distillation, we use mean average precision (MAP) as well as three precision-oriented measures: precision at ranks 5 and 10 (P@5, P@10), and mean reciprocal rank (MRR).

We determine statistical significance of differences using a two-tailed paired t-test with $\alpha = .05$. Significant changes are indicated using $^{\blacktriangle}$ (significant increase) or $^{\blacktriangledown}$ (significant decrease).

4.4.3 Smoothing

The performance of language modeling-based retrieval methods is highly responsive to smoothing [268]. To estimate the smoothing parameter λ_{blog} in Eq. 4.5 in our model, we set λ_{blog} equal to $\frac{n(blog)}{\beta+n(blog)}$, where $n(blog)$ is the length of the blog (i.e., we sum the lengths of all posts of the blog). Essentially, the amount of smoothing applied to a given

blog model is proportional to the length of that blog (and is like Bayes smoothing with a Dirichlet prior [163]). This approach is consistent with the observation that if a blog contains only few posts, estimation of the blog model is less robust and background probabilities are relatively more reliable and should thus make a larger contribution to the model. We set β to be the average blog length in the test collection (here, $\beta = 17,400$).

4.4.4 Parameter estimation

For the functions *norm* and *qmix* we need to set parameters σ and γ. We performed a sweep over possible (and sensible) values of both parameters ($0 < \sigma < 1.0$; $0 < \gamma < 0.1$) and evaluated the performance on MAP. Based on the results of the sweep, we select $\sigma = .05$ for *norm* and $\gamma = .05$ for *qmix*. Note that we are not trying to optimize the performance by selecting the best parameter, rather, we want to see the impact of the model parameter on the retrieval performance. For this reason, the generalization ability of the parameter setting is not considered.

4.5 Results

Let us revisit our research questions. For our first question, *How do we measure topical consistency for a blog?*, we offer our coherence score as a solution.

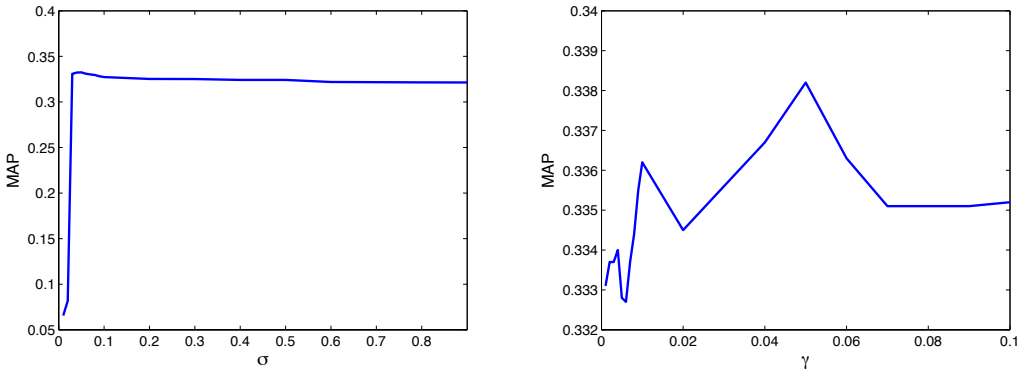
In this section, we turn to the second question, *How can we use the coherence score in our blog retrieval process?* A number of options, ranging from treating the coherence as a simple prior to modeling it as a multiplicative factor whose contribution is a function of the *RSV* of a blog, are proposed in Section 4.3. We now compare the results of these options, analyze the outcomes. In Section 4.6, we look into the third research question of *how the sample size influences estimating the coherence score of a blog and what the impact is on blog feed retrieval*.

Table 4.3 lists the results for our baseline model, *baseline*, which uses a uniform prior, our straightforward implementation of coherence, *prior*, which uses $Co(blog)$ (cf. Eq. 3.2) as prior, and the five experimental models, designated *lin*, *norm*, *quad1*, *quad2*, and *qmix* according to which version of the weighted coherence score *wCo* they integrate.

The run using coherence as a prior performs significantly worse than the baseline in terms of MAP, but shows slight (non-significant) improvements on early precision (P@5) and MRR. We can see that all weighting functions show some improvement over the baseline, with *qmix* performing best in terms of MAP and MRR. The improvement gained over the baseline by applying this function as a weight to the coherence score is significant. We can see that the coherence score does not only help MAP and MRR, but also shows improvements in terms of P@5 and P@10 in most cases, although not significant.

Model	MAP	P@5	P@10	MRR
baseline	.3272	.4844	.4844	.6892
prior	.2945▼	.5022	.4822	.6959
lin	.3326	.5022	**.5067**	.7266
norm	.3325▲	.5022	.4822	.7103
quad1	.3327	.5022	**.5067**	.7377
quad2	.3365▲	.5022	.5022	.7154
qmix	**.3382**▲	**.5067**	.5022	**.7394**▲

Table 4.3: Results of coherence score, implemented as prior, and using linear function (*lin*), normal distribution (*norm*), quadratic function 1 (*quad1*), quadratic function 2 (*quad2*), and the combination of quadratic function 1 and 2 (*qmix*). Significance computed against the baseline.



(a) *norm* (b) *qmix*

Figure 4.3: Impact of model parameters on retrieval performance.

As described in Section 4.4.4, we set the parameter values of *norm* and *qmix*, namely σ and γ, by sweeping over a range of possible values. In order to see the impact of these parameters on the end-to-end retrieval performance, in Figure 4.3 we illustrate the relation between the values of the model parameters and the retrieval performance in terms of MAP. We see that for *norm*, with the change of σ, MAP reaches a global maximum at $\sigma = 0.05$, and afterwards, the MAP scores decrease slightly without dramatic changes. For *qmix*, the MAP scores across different values of γ do not differ significantly in general and a global peak at $\gamma = 0.05$ can be found. Note that the y-axis of Figure 4.3(a) and that of Figure 4.3(b) have different ranges. In addition, since the change of MAP scores remains below 0.001 for $\gamma \in [0.1, 0.9]$, in Figure 4.3(b) we only show the MAP scores for $\gamma \in [0.001, 0.1]$ where the change is relatively more obvious.

Let us take a closer look at the results per test query. In Figure 4.4 we compare the performance of each of the models to the baseline and plot the increase or decrease in AP for each query. The plots show that (i) *norm* increases performance in 31 of

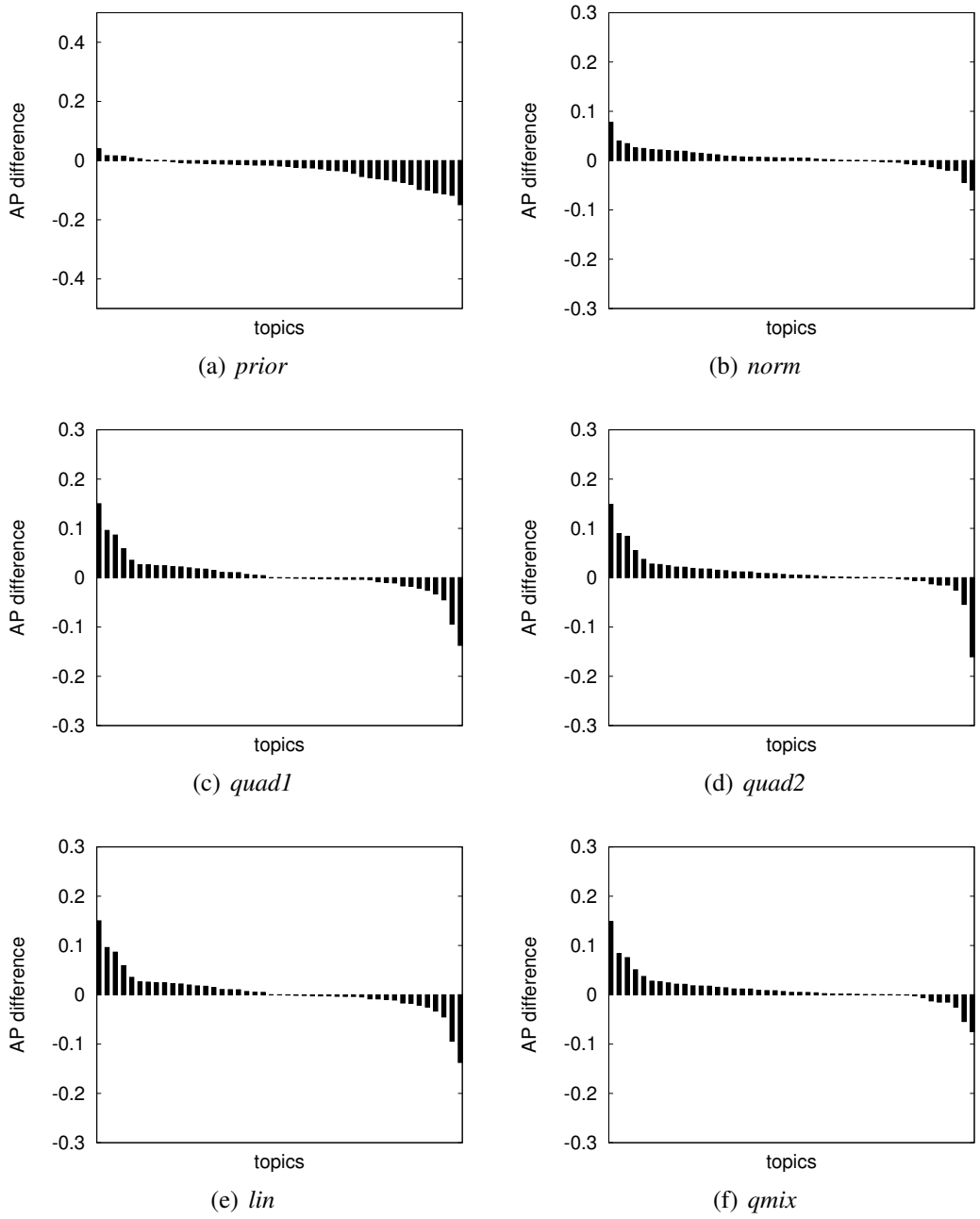
45 queries, but gains are moderate, (ii) *quad1* hurts more queries than it improves (23 vs. 22), (iii) the same goes for *lin* (again 23 vs. 22), (iv) in both cases the maximum increase in AP is high (.15 for query 974), but so is the maximum drop (-.14 for query 979), (v) *quad2* improves performance in 34 of 45 queries, but also shows a large drop for several queries, and finally (vi) *qmix* improves over the baseline in 35 of 45 queries, with a limited drop in AP for the worst performing query (-.07 for query 979). The query that improves most after integrating the coherence score into the model is query 974 (*tennis*), for all models. Query 979 has the worst performance (*lighting*), for all models. Queries whose performance neither improved nor degraded include query 951 (*mutual funds*), query 969 (*planet*), and query 933 (*buffy vampire slayer*). We hypothesize that the potential of the coherence score to improve retrieval performance for a query is (i) related to the breadth of the vocabulary that a blogger uses to discuss the query, (ii) the ability of the query to inspire bloggers over time and (iii) spam blogs whose word distributions cause them to be relevant to that query.

What happens when we explore the per-query differences between the run using coherence as a prior and the runs using the weighting functions? Three queries score worse using weighting functions compared to the prior: 953 (*biofuels may damage forests*), 957 (*Russia*), and 992 (*copyright law*). On the other hand we see three queries that are in the top 3 of most improved queries over the prior run (for all weighting functions): 974 (*tennis*), 973 (*autism*), and 954 (*Mac*). In general, very few queries actually perform better in the prior run than using the weighting functions (8-12 queries out of 45).

Finally, we look at the differences between the runs using the various ways of weighting coherence and see what causes the final evaluation results to be different: Are certain queries hurt by one function, but improved by another? Or do we see a general trend of queries improving or dropping for all functions, just differing in the degree of gain or loss? We try to answer this question using several queries as examples. First, queries 979 and 982 drop most and second most for all functions. At the other end of the spectrum, we have a similar, consistent behavior for queries 974 (improves most), 994 (improves second most), and 995 (improves third most). Only few queries show different behavior: query 964 (*violence in sudan*) improves for *norm*, *qmix*, and *quad2*, but drops for *lin* and *quad1*. Also, query 992 (*copyright law*) drops in all cases, except for *norm*. The overall picture however, shows consistent behavior for queries over all functions, with the level of improvement (or loss) making up for the differences in MAP between the runs.

4.6 Further discussion

Following the assumption we made in our blog retrieval model, for each blog there is a blog model that generates the texts we observe. Since the Blog06 collection is crawled in a certain period, for a given blog we can see it as a sample drawn from an underlying distribution generated according to the blog model. One would expect that the blogs



(a) *prior* (b) *norm*

(c) *quad1* (d) *quad2*

(e) *lin* (f) *qmix*

Figure 4.4: AP differences between baseline and (left-to-right, top-to-bottom) coherence as prior, *norm*, *quad1*, *quad2*, *lin*, and *qmix*.

judged relevant, i.e., blogs having a recurring interest in a given topic, are generated by blog models that generate blog posts with topical consistency. However, since we only see the posts collected during 11 weeks, the true topical distribution of the blog is only approximated by this observed sample.

Intuitively, in order to get a good estimation of the coherence of the underlying topical structure of the blog model, a certain number of posts should be contained in the sample under observation. Also, recall that in Section 3.2 on page 38 we have seen that the size of document sets has an impact on the calculation of coherence score: with same proportion of documents focusing on certain topic, larger document set generates a higher coherence score than smaller document set. This impact is especially significant when the document set is small, for example, less than 15 documents.[4] From Table 4.2 in Section 4.2.2 we see that there exist many "small" blogs, e.g., blogs containing less than 10 posts.

This leads us to the following questions. What is the impact of the sample size on the estimation of the coherence of the true topical structure of the underlying blog model? Can we decide on a minimum number of posts to achieve a reliable estimation? And how would this threshold impact blog feed retrieval performance? Below, we address these questions with exploratory experiments.

4.6.1 Impact of sample size on the estimation of coherence

Intuitively, we expect that a larger sample will provide us with a better approximation of the true topic distribution of the population, i.e., a blog with more posts within the 11 week period of the data set should be a better approximation of the distribution of the topical structure of the blog in an infinite amount of time. Moreover, it is also intuitive that populations of different sizes require different minimum sample sizes for a reliable approximation. Since we do not know the size of the population, i.e., we do not know the number of posts a blog contains outside the 11 weeks covered by the data set, we need to decide on a minimum number of posts that would be sufficient for populations of different sizes.

To this end, we collect blogs with different numbers of posts from the Blog06 collection: blogs with at most 50 posts, 50–100 posts, 399–499 posts, 500–999 posts. For each number of these four groups, we sample 50 blogs for experiments.

For each blog B we collected, we calculate its coherence, which we denote as $Co(B)$. We then sample a different number of posts: 5, 10, 20, 30, 40, and 50,[5] and calculate the coherence score for each sample, which we denote as $Co(S^k)$, where $k = 5, 10, 20, 30, 40, 50$ is the sample size. For each sample size, we generate 30 runs. We analyze how the value of $Co(S^k)$ approximates the value of $Co(B)$ as k changes by calculating the Mean Squared Error (MSE) of the sample coherence scores from the real coherence scores derived from the original blog using Eq. 4.15.

$$MSE(Co(S^k)) = \frac{1}{n}\sum_i \Big(Co(s_i^k) - Co(B)\Big)^2, \tag{4.15}$$

[4]Note that this observation is made under simplified assumptions, i.e., the situation where the document sets can be divided into two self-coherent and mutually exclusive subsets. Nevertheless, it gives us sufficient motivation to check this phenomenon in practice.

[5]For the set of blogs of 50 posts, we ignore the case of sampling 50 posts, as in this case it is equivalent to select all posts in a blog and therefore no approximation is needed.

where $i = 1, \ldots, 30$, $S^k = \{s_i^k\}_{i=1}^{30}$ is the set of samples from the 30 runs, which are drawn from the original blog B.

To summarize the trends of the impact of sample size on estimating the real coherence for a blog, we take the average MSE of the 50 blogs of different number of posts. Figure 4.5 shows the results. We see that as the sample size increases, the average MSE decreases. In particular, after 20 posts, the changes in average MSEs become very small compared to that before 20 posts. This trend applies to blogs with different numbers of posts, which suggests that no matter how large the actual size of the blog would be in an infinite amount of time, a minimum number of 20 posts can achieve a stable estimation of the true topical structure of a blog.

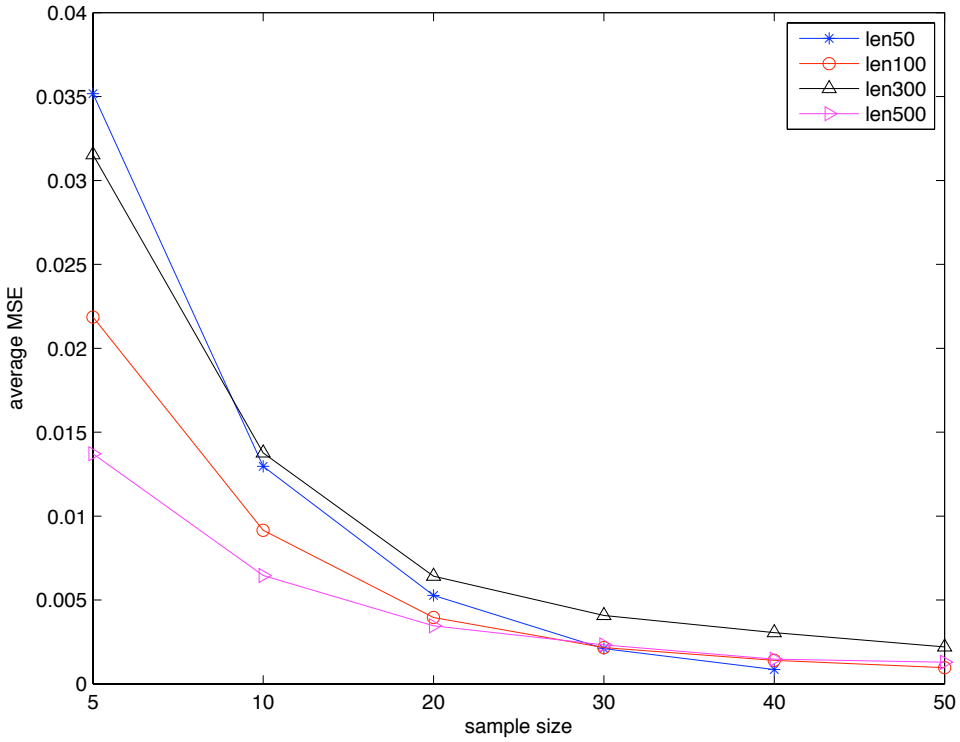


Figure 4.5: Relation between the sample size and the average MSE of the sampled coherence score from the real coherence score. Here, len50, len100, len300, len500 denote the samples of blogs with 50, 100, 399–499, 500–999 blogs, respectively.

4.6.2 Relation between the population coherence and the accuracy of being approximated by sampled coherence

One may notice that in Figure 4.5, for the same sample size, e.g., 5 posts sample, blogs with 500–999 posts have a lower MSE than blogs with 50 posts. This is counterintu-

blogs of different sizes	50	100	300–499	500–999
population coherence score	0.5344	0.5755	0.5091	0.7339

Table 4.4: The average coherence score of blogs with different number of posts.

itive. Indeed, we would expect that it is more difficult to approximate the distribution of a large population than a small one with the same amount of samples. In other words, we expect the average MSE of blogs with more than 500 posts to be higher than that of blogs with 50 posts. This unexpected phenomenon suggests that there are other factors besides the sample size that impact the estimation of the topical structure of the underlying blog model. A potential dimension is the coherence of the original blog, i.e., the coherence score of the population.

In Figure 4.6, we fix the sample size, and show the relation between the MSE of the sampled coherence score and the population coherence score. We see that the relation is non-linear, but there exists a pattern, which can be approximated by a quadratic function (shown in the plots). Particularly, if the population is extremely coherent, or extremely random, it has a better approximation.

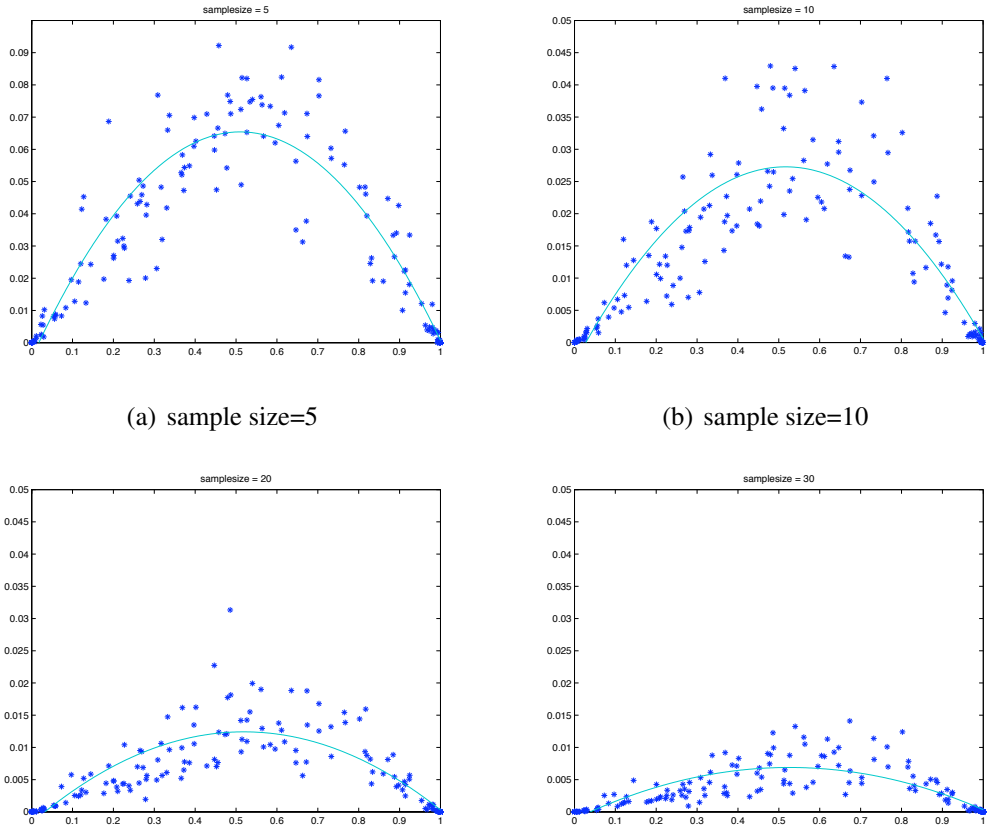
In Table 4.4, we list the average coherence score of blogs with different numbers of posts that we used in the experiment discussed in Section 4.6.1. As we can see, the average population coherence score of blogs with more than 500 posts is much higher than that of blogs with 50 posts. This explains the phenomenon shown in Figure 4.5.

To wrap-up, the experiment in this section shows that for a given posts sample size, the coherence of the population is a factor that impacts the accuracy of the approximation. Populations with extremely random or extremely coherent topical structures are easier to be approximated. The relation between the population coherence and the accuracy of being estimated by sampled coherence is non-linear but does have a pattern (i.e., close to a quadratic relation).

4.6.3 Impact of sample size on blog feed retrieval

Exploring the impact of sampling size a step further, we experiment with post thresholds in the retrieval process. Blogs with fewer posts than the threshold are discarded from the results (both in the baseline setting, as well as in the coherence-based runs), leaving us with a thresholded blog feed retrieval runs. We use thresholds between 0 and 50 posts, and use the best performing parameter setting for the five models (i.e., $\sigma = .05$ for *norm* and $\gamma = .05$ for *qmix*). Figure 4.7 plots the relative increase in MAP for each of the models over the baseline for different thresholds.

From the plot we can conclude that the greatest relative improvement over the baseline occurs when only blogs with more than 20 posts are taken into consideration. The function *norm* is the only one to have its peak at a threshold of 30 posts. On the other hand, if the threshold eliminates too many blogs, the relative improvement will decrease since there may be very few relevant blogs left after thresholding. Table 4.5 lists the results for each of the functions and the baseline when using a threshold of



(a) sample size=5

(b) sample size=10

(c) sample size=20

(d) sample size=30

Figure 4.6: Relation between the population coherence score (X-axis) and the MSE of the sampled coherence score (Y-axis).

20 posts. The results show that in three cases the improvement over the baseline is significant (in terms of MAP), and that, again, the weighting function *qmix* performs best on all metrics.

The experiments in Sections 4.6.1 and 4.6.3 lead to the conclusion that coherence becomes beneficial for blogs when a blog contains more than 20 posts. This result suggests that it would be worth looking into the development of methods to estimate priors for blogs that are (currently) too short to derive benefit from the coherence score.

4.7 Conclusion

In this chapter we proposed a method to counteract the effects of topical noise in blogs with the goal of performing blog feed retrieval. For a blog to be relevant in a feed search task, it should show recurring interest in a given topic, something that is hard

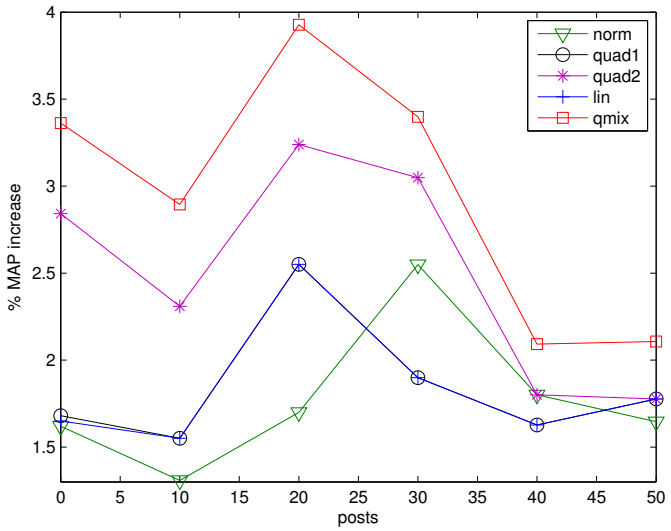


Figure 4.7: Effect of threshold on difference in MAP between models and baseline.

function	MAP	P@5	P@10	MRR
baseline	.2470	.4578	.4511	.6930
lin	.2533	.4756	.4689	.7174
norm	.2512▲	.4756	.4622	.7030
quad1	.2534	.4756	.4689	.7285
quad2	.2550▲	.4756	**.4711**	.7061
qmix	**.2567**▲	**.4800**	**.4711**	**.7321**

Table 4.5: Results of weighted coherence score applied to blogs with a minimum of 20 posts. Significance computed against the baseline. (Note that, compared to Table 4.3, the baseline has changed, due to the fact that blogs with fewer than 20 posts are eliminated from the collection.)

to measure due to the noisiness of blogs on a blog level. Within this context, we raise three research questions:

RQ2a. How do we measure topical consistency for a blog?

RQ2b. How can we use the coherence score in our blog retrieval process?

RQ2c Given that the collection we use in our experiments only provides us with a sample of blog posts generated by the underlying blog models, how does the sample size influence the estimation of the coherence and how does this influence blog feed retrieval?

For the first research question, we argued that established cohesion measures, in particular lexical cohesion calculated on the basis of lexical chains, are not suited for mea-

suring topical consistency in the blogsphere and proposed our coherence score which captures the topical clustering structure of a set of documents relative to a background collection. The coherence score can be calculated relatively efficiently. The calculation makes use of collection statistics only and requires neither external resources nor collection-specific parameter optimization. Applied to blogs, the coherence score reflects topical consistency, in other words, the level of topical noise of a blog.

With respect to research question RQ2b, we find that incorporating the coherence score in our retrieval framework required us to look at the relation between coherence and relevance. In case of a (topically) relevant blog, this blog should not be highly favored in the final ranking unless it is also topically coherent. On the other hand, blogs that have high topical coherence because they consistently treat a different topic than the given topic, should not enjoy unjustified promotion within the final ranking. To prevent this, we proposed weighting the coherence score by a notion of topical relevance. We compared two methods of incorporating the coherence: (i) a query-independent method, using coherence as prior, and (ii) a relevance-dependent method, where the coherence is weighted using a function of the retrieval score. Results show that the second method outperforms the baseline model, while the first method does not. Furthermore, the *qmix* function performs best with significant improvement over the baseline on MAP and MRR, and non-significant improvements on the other metrics.

For research question RQ2c, following the intuition that the posts in our data set are a sample of the blogger's posts, we expected a larger sample size to be a better approximation of the true distribution of posts. Our analysis of the relation between the sample size and the average deviation of the sampled coherence from the actual coherence of a blog shows that from 20 posts onwards this deviation does not change much anymore, indicating that 20 posts is the minimum sample size needed to get a proper estimation. This is further supported by blog feed retrieval experiments using only blogs that have more posts than a given threshold: using a threshold of 20 posts shows maximum relative improvement over the baseline.

We have shown the coherence score to be effective in capturing topical consistency in user generated content. Future work will focus on further optimization of the coherence score for use in blog feed retrieval, involving, for example, in-depth investigation of query-specific performance that could lead to further refinement of the weighting functions. An extension of the coherence score to other areas of user generated content, such as user reviews or audio blogs (podcasts) is a further avenue of future research.

Chapter 5

Using Coherence-Based Score for Query Difficulty Prediction

Robustness is an important feature of information retrieval (IR) systems [250]. A robust system achieves solid performance across the board and does not display marked sensitivity to difficult queries. IR systems stand to benefit if, prior to performing retrieval, they can be provided with information about problems associated with particular queries [85]. Work devoted to predicting query difficulty (also called query performance) [7, 33, 51, 52, 90, 93, 100, 265, 272] is pursued with the aim of providing systems with the information necessary to adapt retrieval strategies to problematic queries. For a survey on the work on this research topic, a recent book by Carmel and Yom-Tov [33] covers a wide range of query performance predictors proposed in the literature. Moreover, Hauff [90] has conducted extensive comparative studies on various types of predictors in her thesis, including the predictors we discuss in this chapter.

In this chapter, we investigate the usefulness of the coherence score in predicting query difficulty in a *pre-retrieval* setting. Specifically, we ask the following research questions:

RQ3a. Can we use the coherence score to measure query ambiguity?

RQ3b. Can we use query ambiguity as measured by coherence-based scores to predict query performance in an ad-hoc retrieval setting?

We posit that the performance of a query is correlated with its level of ambiguity. That is, we assume that the user's information need is specific and clearly defined, and therefore a query tends to retrieve non-relevant documents when it is ambiguous. For example, when a user searches for information about "java program," the query "java" may retrieve documents on topics such as "java island" or "java coffee." Here, the retrieval performance of a query is influenced by two factors. The first factor is the query itself. In the above example, if a query is associated with multiple subtopics or interpretations, it is likely that some of the subtopics or interpretations are non-relevant. Second, the performance for a given query also depends on the document collection we

use: an ambiguous query only affects retrieval performance if the collection contains documents associated with non-relevant interpretations of the query.

The *query coherence scores* we propose are designed to reflect the quality of individual aspects of the query, following the suggestion that "the presence or absence of topic aspects in retrieved documents" is the predominant cause of current system failure [85]. We use document sets associated with individual query terms to assess the quality of query topic aspects (i.e., subtopics), noting that a similar assumption proved fruitful in [265]. We consider that a document set associated with a query term reflects a high-quality (i.e., non-ambiguous) query topic aspect when it is: (1) topically constrained or specific and (2) characterized by a clustering structure tighter than that of some background document collection. These two characteristics are captured by coherence and for this reason we chose to investigate the potential of coherence-based scores. Like the clarity score [51, 52], our approach attempts to capture the difference between the language usage associated with the query and the language usage in the background collection.

We propose three query coherence scores. The first query coherence score, QC-1, is an average of the coherence contribution of each query word and only has the effect of requiring that all query terms be associated with high-quality topic aspects. This score is simple and efficient. However, it does not require any semantic overlap between the contributions of the query words. A query topic composed of high-quality aspects would receive a non-zero QC-1 score even if those aspects were never reflected *together* in a document. Hence, we further develop two alternative scores that impose the requirement that, in addition to being associated with high-quality topic aspects, query words must be topically close. The second query coherence score, QC-2, adds a global constraint to QC-1. It requires the union of the set of documents associated with each query word to be coherent. The third score, QC-3, adds a proximity constraint to QC-1. It requires the document sets associated with individual query words be close to each other.

The next section further explains our coherence-based scores. After that we describe our experiments and results. We conclude with a discussion and outlook.

5.1 Query coherence scores

Given a document collection C and query $Q = \{q_i\}_{i=1}^{N}$, where q_i is a query term. We define R_{q_i} as the set of documents associated with a query word, i.e., the set of documents that contain at least one occurrence of the query word. The coherence of R_{q_i} reflects the quality of the aspect of a query topic that is associated with query word q_i. The overall query coherence score of a query is based on a combination of the set coherence scores contributed by each individual query word. Below, we first discuss coherence on a set of documents and then present our three query coherence scores.

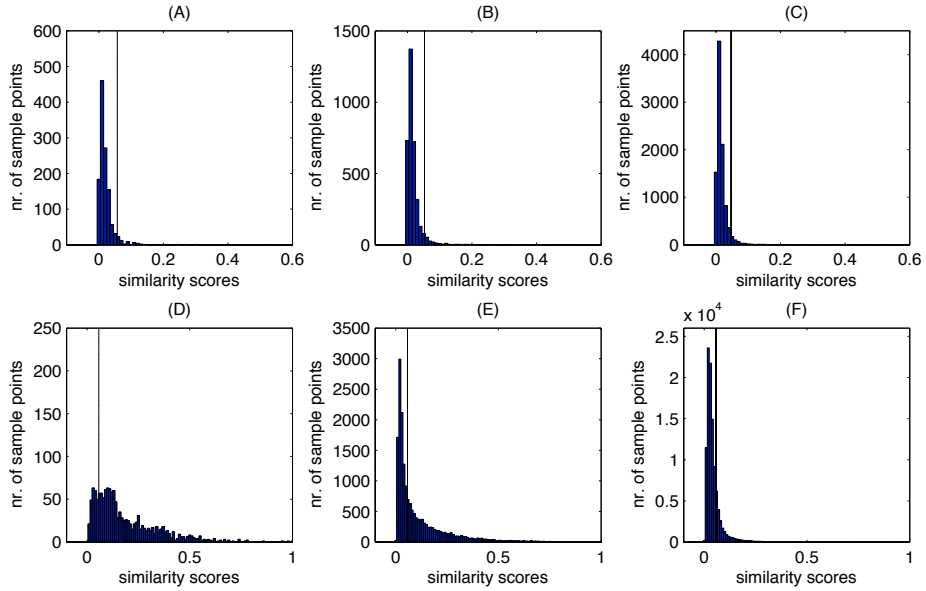


Figure 5.1: Distribution of document similarities from subsets of TREC AP89+88 (as introduced in Section 3.3.1 on page 40). (A)–(C) Randomly sampled 50, 100, and 500 documents, respectively; (D) R_Q determined by query21, $Co(R_{Q21}) = 0.8483$; AP(Q21)=0.1328; (E) R_Q determined by query57, $Co(R_{Q57}) = 0.7216$; AP(Q57)=0.0472; (F) R determined by query75, $Co(R_{Q75}) = 0.2504$; AP(Q75)=0.0027.

5.1.1 The coherence of a set of documents

As defined in Chapter 3, the coherence score is a measure for the relative tightness of the clustering of a specific set of data with respect to the background collection. In a random subset drawn from a document collection, few pairs of documents have high similarities. In Figure 5.1 we illustrate coherence of documents collected in different ways. The coherence of each document set is calculated as defined in Eq. 3.2 on page 36. Plots A, B, and C in Figure 5.1 show that pairs having similarity scores higher than the threshold τ (the vertical line) are proportionally rare cases in a random sample, independent of sample size. Plots D, E and F show the distribution of document similarities for a collection subset associated with a one-word query, which we use to illustrate the properties of the R_{q_i}, the collection subset associated with a single query word q_i. Plots D, E, and F are ordered by decreasing coherence score, which corresponds to an increasing proportion of dissimilar document pairs. Plot F approaches the distribution of the random samples from the background collection. Initial support for the legitimacy of our approach is derived from the fact that across these three queries decreasing set coherence of R_{q_i} corresponds to decreasing AP (as introduced in Section 2.4.2 on page 27).

5.1.2 Scoring queries based on coherence

For a given query $Q = \{q_i\}_{i=1}^{N}$, we propose three types of query coherence score. The first requires that each query word have a high contribution to the coherence of the query. This score reflects the overall quality of all the aspects of a topic.

QC-1 Average query term coherence:

$$QC\text{-}1(Q) = \frac{1}{N}\sum_{i=1}^{N} Co(R_{q_i}), \tag{5.1}$$

where $Co(R_{q_i})$ is the coherence score of the set R_{q_i} determined by the query word q_i. This score is simple, but leaves open the question of whether query aspects must also be semantically related. Therefore, we investigate whether QC-1 can be improved by adding constraints that would force the R_{q_i}'s to be semantically related. The second query coherence score adds a constraint on global coherence, multiplying QC-1 by the coherence of $R_Q = \bigcup_{i=1}^{N} R_{q_i}$.

QC-2 Average query term coherence with global constraint:

$$QC\text{-}2(Q) = Co(R_Q)\frac{1}{N}\sum_{i=1}^{N} Co(R_{q_i}). \tag{5.2}$$

The third query coherence score adds a constraint on the proximity of the R_{q_i}'s, multiplying QC-1 by the average of the closeness of the centers of the R_{q_i}'s.

QC-3 Average query term coherence with proximity constraint:

$$QC\text{-}3(Q) = \frac{S}{N}\sum_{i=1}^{N} Co(R_{q_i}) \tag{5.3}$$

$$S = \frac{\sum_{l\neq k}^{N} Similarity(c(q_k), c(q_l))}{N(N-1)}, \tag{5.4}$$

where S is the mean similarity score of each pair of cluster centers $c(q_i)$ of the R_{q_i}'s. Here, $c(q_i)$ is calculated as

$$c(q_i) = \frac{1}{M}\sum_{d\in R_{q_i}} \vec{d}, \tag{5.5}$$

where M is the total number of documents contained in R_{q_i} and $\vec{d}$ is a document in R_{q_i} represented using a vector space model.

Below, we compare the performance of the three query coherence scores.

5.2 Evaluation

5.2.1 Experimental setup

We run experiments to analyze the correlation between the proposed query coherence scores and the retrieval performance. Following [51], TREC datasets AP88+89 (as

introduced in Section 3.3.1) are selected as our document collection. We use TREC topics 1–200 with the "title" field. We experiment with a number of retrieval models, including BM25 [202], TFIDF [203, 234], and the DFR model with the PL2 and the DH13 weighting schemes [6]. We use the Terrier [186] implementation of these models with default parameter settings.

We calculate the coherence score for a document set associated with a query term as defined in Eq. 3.2 on page 36. The threshold τ is determined as described in Section 3.1.1 on page 36, and cosine similarity is used as the measure of similarity between documents. For large sets R (e.g., $> 10,000$ documents), we approximate the coherence score by using the "collection" score (the threshold τ); we posit that a set R with many documents has a coherence score similar to the collection.

5.2.2 Evaluation measure

We use Spearman's ρ to measure the rank correlation between the coherence score and the Average Precision. The higher this correlation, the more effective the scoring method is in terms of predicting query difficulty. Different retrieval models are applied so as to show stability of our observations across models.

5.2.3 Results

Table 5.1 shows that all three coherence scores have a significant correlation with AP. In general, QC-2 and QC-3 show a higher positive correlation with the AP than QC-1. However, their predictive ability is not substantially stronger than QC-1, judging from the difference between the correlation coefficients of QC-1 and that of QC-2 and QC-3, though they do take the semantic relation between query words into account. Since the coherence score is the proportion of "coherent pairs" among all the pairs of data points, and the similarity score can be pre-calculated without seeing any queries, the run-time operation for QC-1 is a simple counting of the "coherent pairs." The same holds for QC-2, but with more effort for the extra term R_Q. Both are much easier to compute than QC-3, which requires the calculation of the centers of the R_{q_i}'s that need to be processed at run-time. Therefore, taking into account its computational efficiency and the limited improvements seen in the alternative QC's, QC-1 is the preferred score. Moreover, even thought it is a pre-retrieval predictor, QC-1 has a competitive prediction ability compared to other post-retrieval methods such as the clarity score [51]; see Table 5.2.

5.2.4 Hauff's experiments

In addition to our preliminary experiments, Hauff [90] has conducted further experiments in analyzing the performance of QC-1 and QC-2. In Hauff's experiments, the coherence score is implemented as described in [100] with the same experimental settings as described here. Additional TREC test collections (TREC 4+5 [248],

	QC-1		QC-2		QC-3	
Model	ρ	p-value	ρ	p-value	ρ	p-value
BM25	0.3295	1.8897e-06	0.3389	0.0920e-05	0.3813	2.5509e-08
DLH13	0.2949	2.2462e-05	0.3096	0.8180e-05	0.3531	2.9097e-07
PL2	0.3024	1.3501e-05	0.3135	0.6167e-05	0.3608	1.5317e-07
TFIDF	0.2594	2.0842e-04	0.3301	0.1805e-05	0.3749	4.5006e-08

Table 5.1: The Spearman's rank correlation of query coherence scores with average precision. Queries: TREC topics 1–200; document collection: AP89+88.

Score	CS	QC-1	QC-2	QC-3
ρ	0.368	0.3443	0.3625	0.3222
p-value	1.2e-04	4.5171e-04	2.1075e-04	0.0011

Table 5.2: The Spearman's rank correlation of clarity score (CS) and query coherence score (QC) with AP: the correlation coefficient ρ and its corresponding p-value. The queries are TREC topics 101–200, using title only. AP values obtained by running BM25; the clarity scores of column 1 are taken from [51].

WT10g [232] and Gov2 [39]) are used. TREC4+5 is similar to AP89+88: it is relatively small compared to the other two collections, consisting of news articles. WT10g and Gov2 are large collections consisting of Web crawls. TFIDF, BM25 and Language Model (LM) are included as retrieval models.

The conclusion of Hauff's experiments can be summarized as follows.

- First, the performance of QC-1 and QC-2 varies across collections and the better performance is achieved on smaller collections. Best performance is achieved on TREC 4+5, where both predictors show relatively stable positive correlations across all three retrieval models and significant results are achieved. The performance on WT10g and Gov2 are not as stable, in many cases only insignificant correlations are found between our predictors (i.e., QC-1 and QC-2) and the AP.

- Second, when LM is used as retrieval model, the rank correlation between the query coherence scores and AP increases with an increasing amount of smoothing.

Combining the observations made from our experiments and those of Hauff's experiments, one important conclusion here is that the query coherence scores are more effective on small collections (particularly, on a specific domain such as news) than on large and Web based collections. One possible explanation is that, in smaller collections, especially in a single domain such as news articles where the language usage is often more confined compared to that on the Web, the query term ambiguity is captured well by the topical coherence of documents associated with it. In a Web collection, however, *every* query term may be associated with more diverse documents, including spam, which may reduce the distinction between non-ambiguous terms and ambiguous

terms. Particularly, when using the heuristic approximation for large document sets associated with a query term (as described in 5.2.1 on page 72), in large collections, it is very likely that this approximation is used for most of the queries, as most of the queries may be associated with a document set with more than 10,000 documents.

5.3 Discussion and conclusions

With respect to our two research questions RQ3a and RQ3b as stated on page 69, we have the following answers. We introduced coherence-based measures for query difficulty prediction. The coherence score of the set of documents associated with a single query term is used as a measure of the quality (i.e., level of non-ambiguity) of the query term. We then experimented with three ways to combine the coherence scores of each query term into a single score as performance predictors for a query. Our initial experiments on short queries show that the coherence score has a strong positive correlation with average precision, which reflects the predictive ability of the proposed score.

Hauff's experiments, on the other hand, have raised further open issues for these predictors. For example, what makes our predictors less effective on large collections? how do we measure query ambiguity on large collections such as the Web? Further, with respect to Web retrieval, it is an open question whether query ambiguity is an important factor responsible for query performance. For example, strategies such as result diversification are often used to deal with ambiguous or multi-faceted queries. That is, without knowing the actual user's information need, the retrieval system presents a list of documents covering as many as possible subtopics associated with the query. Within this specific task scenario, the importance of query ambiguity with respect to the query performance may need to be reconsidered, which we leave as future work.

Conclusion to Part I

In Part I of the thesis, we addressed the research theme *topical coherence*. We studied two major issues with respect to this research theme: (i) how do we measure the topical coherence of a set of documents? and (ii) how do we use the proposed coherence measure in IR tasks where such a measure is needed?

In Chapter 3 we proposed a coherence score that measures the topical coherence of a set of documents by evaluating the relative tightness of the clustering structure of the document set as compared to a background collection. Empirical evaluation on simulated text data shows that the coherence score is effective in capturing the topical coherence of a document set.

Further, in Chapter 4 and 5 we applied the coherence score to two IR tasks, namely, blog feed retrieval and query performance prediction. In both cases, the coherence score was shown to be useful. In the case of blog feed retrieval, we use the coherence score as a measure of the topical consistency of the blog posts belonging to the same blog. By incorporating the coherence score into a language modeling based blog retrieval model, we achieved significant improvements in retrieval performance on the blog feed retrieval task. For query performance prediction, we posit that the performance of a query in an ad-hoc retrieval setting is related to the level of ambiguity of the query. We use coherence score of a set of documents associated with a query term as a measure of the level of ambiguity of the query term. Then, the per-query coherence scores are aggregated into a single score which is an indication of the performance of a query. Empirical results on a news collection show that the coherence-based predictor has a significant positive correlation with the performance of queries in terms of average precision.

Given the effectiveness of the coherence score in capturing the topical coherence of a document set shown in these tasks, in the following chapters, we will occasionally use the coherence score in situations where such a measure is needed.

Part II

Relevance, Diversity and the Cluster Hypothesis

Chapter 6

Diversity and the Cluster Hypothesis

In Chapter 5, we used coherence-based scores to predict the ambiguity (or lack of coherence) of a query. While our empirical results show that there is a statistically significant correlation between the query coherence and the query performance in retrieving topically relevant documents in an ad-hoc retrieval setting, some remarks should be made about the experimental settings and results. First, the prediction is made on the assumption that the user's information need is specific and clearly defined. Or in other words, we assume that users know what they are looking for. This is also reflected by the test collection we use, where the queries contain "descriptions" and "narratives" that describe the information need in a specific and detailed manner and the assessors make their relevance judgements accordingly. In practice, users may not be aware or do not possess sufficient knowledge to be able to formulate such a specific information need. For example, a user may have heard a new expression and explores its meaning on the Internet, while the expression may refer to multiple senses (ambiguous) or it may cover a wide range of subtopics (multi-faceted). Second, the prediction of the coherence of a query does not automatically imply a solution to deal with queries that are ambiguous or multi-faceted.

In Part II of the thesis, we discuss one strategy that deals with ambiguous or multi-faceted queries, that is, the strategy of *result diversification*. We investigate the role of topic structure in this particular scenario from the perspective of the cluster hypothesis, the hypothesis that relates the topic structure of a collection with the (topical) relevance of documents in response to an information request (recall the introduction of the cluster hypothesis in Section 2.2.1 on page 18).

In this chapter, we re-visit the cluster hypothesis in the context of result diversification in seeking answers to the research question below:

RQ4. *How do we interpret the cluster hypothesis in the context of result diversification?*

In the next chapter, we propose a cluster-based result diversification framework which aims at answering the following research question:

RQ5. *Can query specific clustering be used to improve the effectiveness of result diversification?*

6.1 Introduction

Queries submitted to web search engines are often ambiguous or multi-faceted in the sense that they have multiple interpretations or subtopics [3]. For ambiguous queries, a typical example is the query "jaguar" that can refer to several interpretations including a kind of animal, a car brand, a type of cocktail, an operating system, etc. Multi-faceted queries are even more commonly seen in practice; for example, for the interpretation "jaguar car" of the query "jaguar," a wide range of subtopics may be covered: models, prices, history of the company, etc. For such queries we often cannot be certain what the searcher's underlying information need is because of a lack of context. One retrieval strategy that attempts to cater for multiple interpretations of an ambiguous or multi-faceted query is to *diversify* the search results [25, 77]. Without explicit or implicit user feedback or history, the retrieval system makes an educated guess as to the possible facets of the query and presents as diverse a result list as possible by including documents pertaining to different facets of the query within the top ranked documents.

Diversification in the manner just described seems at odds with the *cluster hypothesis*, the assumption that *relevant documents tend to be more similar to each other than to non-relevant documents* [105, 245].[1] Intuitively, diversification and cluster-based ranking build on different assumptions about a user's information need. Promoting (all documents within) a single cluster to the top of a ranked list is bound to hurt in terms of diversity-based metrics. But a diversification strategy based on presenting users with samples from multiple clusters amongst the top-ranked documents may promote non-relevant results, thus hurting relevance-based metrics.

In this chapter, we revisit the cluster hypothesis in the context of result diversification and investigate the relation between relevance and diversification. We start with the following research question concerning the cluster hypothesis:

RQ4a *Given a query that is ambiguous or multi-faceted, i.e., associated with several subtopics, do the relevant documents tend to be more similar to each other than to non-relevant documents? Particularly, do ambiguous or multi-faceted queries show different patterns in terms of inter-document similarities compared to specific or single-faceted queries?*

Although the cluster hypothesis does not limit itself to queries that are not ambiguous or multi-faceted, it is not a priori obvious that documents relevant to "jaguar" the car brand would be similar to documents that are relevant to "jaguar" the animal, given that both interpretations are topically relevant to the query "jaguar." On the other hand,

[1] As we have discussed in Chapter 2, the cluster hypothesis has been phrased in different but closely related ways in the literature, here we follow the statement by Hearst and Pedersen [105], see Section 2.2 on page 18 for more details.

compared to ambiguous queries, the cluster hypothesis is more likely to hold in the case of multi-faceted queries, as documents relevant to each facet of a query may be similar to each other due to the fact that they are all associated with a topic with a broader range. In order to distinguish these two situations, i.e., ambiguous versus multi-faceted queries, we ask a follow up research question:

RQ4b. *Do ambiguous queries show different patterns in terms of inter-document similarities from multi-faceted queries?*

Moreover, the distinction between multi-faceted and ambiguous queries is not always as clear-cut as in the "jaguar" example. We will use the categorization, i.e., multi-faceted versus ambiguous, as implemented in the TREC 2009 Web Track diversity task; see below.

In addition, from previous work on query-specific clustering, as discussed in Section 2.2 on page 18, we know that the main reason why query-specific clustering can improve early precision is that among the document clusters, there exist a few *high quality* clusters such that most of the relevant documents are contained in these clusters [105, 140]. Working on TREC-3 data [84], Hearst and Pedersen [105] show that if for each query one clusters the top ranked documents into five clusters, then *"the top-ranked cluster always contains over 50% of the relevant documents retrieved, ... The third, fourth and fifth-ranked clusters usually contain 10% or fewer."* If documents from those high quality clusters are placed at the top of a ranked list, it is very likely that many of the relevant documents are promoted to the top of the ranked list, hence improving early precision. Now consider queries that are ambiguous or multi-faceted, we are interested to see if the same goal can be achieved, specifically,

RQ4c. *Can we cluster the documents retrieved in response to an ambiguous or multi-faceted query in such a way that most relevant documents are contained in a small set of high quality clusters?*

If the answer to RQ4c is "Yes," such a clustering structure is interesting for result diversification. If non-relevant documents can be "clustered away" from the relevant documents, intuitively, the performance of result diversification can be effectively improved by restricting the diversification procedure to documents that are potentially relevant.

We explore answers to our research questions using empirical methods on web data that has been made available through the TREC 2009 Web track [40]. Several features make this test collection appealing for our task, including the size of the document collection and the fact that the queries are derived from query logs. The most important feature, however, is that the track launched a dedicated diversity task that provides queries as well as relevance judgements that are specifically designed for measuring the performance of retrieval systems in terms of diversity. See Section 6.3.1 for details about this test collection.

In the next section, we introduce the empirical methods we use to examine the cluster hypothesis with respect to our research questions. We then specify our experimental

setup in Section 6.3. After that we discuss the results in Section 6.4 and conclude with our answers the research questions in Section 6.5.

6.2 Methods

In this section, we introduce the experimental analysis we set up in order to seek answers to the research questions raised in the previous section. We start by introducing the notation we employ in the reminder of this chapter. Note that Chapter 7 will continue using the same notation.

6.2.1 Notation

Let d, q and D denote a document, query and set of documents, respectively. Given q, we write D_q^R and D_q^{NR} to refer to the explicitly judged relevant documents for q and the explicitly judged non-relevant documents for q, respectively. We write D_q^n for the top n documents retrieved in response to q. In D_q^n we identify a set of K clusters, $C = \{c_k\}_{k=1}^K$. We use the notation $d \in c_k$ to denote the assignment of document d to cluster c_k and write $D_q^{c_k}$ for the set of documents that belong to cluster c_k.

6.2.2 Revisiting the cluster hypothesis

Recall our first research question

RQ4a *Given a query that is ambiguous or multi-faceted, i.e., associated with several subtopics, do the relevant documents tend to be more similar to each other than to non-relevant documents? Particularly, do ambiguous or multi-faceted queries show different patterns in terms of inter-document similarities compared to specific or single-faceted queries?*

The empirical answer to this question can be arrived at easily, by comparing the distribution of similarity scores between relevant documents with that of relevant and non-relevant documents [121, 245]. In order to analyze the difference of inter-document similarity distributions of ambiguous/multi-faceted queries to that of specific or single faceted queries, for each query, we consider the judged relevant documents associated with each subtopic of the query.[2]

In addition to this comparison, we are also interested in a more insightful analysis of the clustering structure present in the relevant documents, i.e., the degree to which the documents focus on a certain topic or topics, and to compare this cluster structure both to the non-relevant documents and to the union of the relevant and non-relevant documents.

[2]The queries we use in our experiments contain explicitly defined subtopics. See Section 6.3.1 on page 87 for description of our test collection.

Specifically, we consider the following data sets associated with a given query q: D_q^R, $D_{q_s}^R$, $D_q^{R\times NR}$, and D_q^{R+NR}, where $D_{q_s}^R$ is the documents judged relevant to a specific subtopic s of query q, and D_q^{R+NR} is the union of all judged relevant and non-relevant documents with respect to q. The notion of $D_q^{R\times NR}$ is explained as follows. The document set contains both D_q^R and D_q^{NR}, i.e., the same amount of documents as in D_q^{R+NR}. However, we only consider the similarity between the document pairs $< d_i, d_j >$, where $d_i \in D_q^R$ and $d_j \in D_q^{NR}$.

We expect that these four sets of documents display different clustering structures in terms of the amount of clusters and tightness of the clusters. Intuitively, we expect that relevant documents associated with a single facet of a query are similar to each other, while relevant documents associated with different facets are not necessarily similar. E.g., documents about "jaguar car" and about "jaguar cat" may both be judged as relevant to the query "jaguar," but tend to appear in different clusters as they are about different interpretations of the query. However, if the cluster hypothesis holds with respect to each single facet, those relevant documents should still show dissimilarity to non-relevant documents. Hence, for $D_{q_s}^R$ we expect that it forms a single tight cluster, and for D_q^R, we expect that it contains a few tight clusters, corresponding to different facets. For $D_q^{R\times NR}$, we expect that some document pairs show certain similarity due to the fact that they are all retrieved by the same query, but in general lack of highly similar pairs. For D_q^{R+NR}, we expect that it has a structure similar to that of $D_q^{R\times NR}$, while mixed with certain clusters consisting of relevant documents. Figure 6.1 provides a rough illustration of the different clustering structures described above: three tight relevant clusters are visible and a loose non-relevant one.

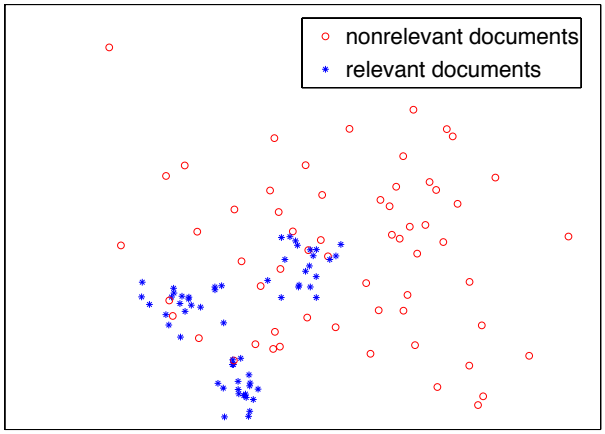


Figure 6.1: An illustration of different clustering structures.

To measure the clustering structure within a set of documents, we summarize the similarity distribution of the document set using the coherence score as introduced in Chapter 3. By comparing coherence scores, we obtain a high-level view of the clustering structure within the four types of document sets, i.e., relevant documents associated

with multiple facets of a query, relevant documents associated with a single facet of a query, non-relevant documents and their union.

6.2.3 Ambiguous versus multi-faceted queries

In order to answer the research question RQ4b:

RQ4b. *Do ambiguous queries and multi-faceted queries show different patterns in terms of inter-document similarities with respect to the cluster hypothesis?*

queries are categorized into two types: ambiguous and multi-faceted.[3] Similar to the previous experiment described in Section 6.2.2, we summarize the inter-document similarity distributions of judged relevant documents associated with a query q using the coherence score. For each query q, we calculate a coherence score $Co(D_q^R)$. We then construct a box-plot to visualize and compare the distributions of coherence scores for the two types of queries.

6.2.4 Distribution of relevant documents among query-specific clusters

We then proceed to introduce the analysis to be used for the third research question:

RQ4c. *Can we cluster the documents retrieved in response to an ambiguous or multi-faceted query in such a way that most relevant documents are contained in a small set of high quality clusters?*

Specifically, for each query, we cluster the documents D_q^N and summarize the distribution of relevant documents over clusters across queries. Following [105], for each query we rank the clusters in descending order with respect to the percentage of relevant documents they contain, that is, with respect to

$$recall(D_q^R, c) = \frac{|D_q^R \cap D_q^{c_k}|}{|D_q^R|}. \tag{6.1}$$

We then construct box-plots for the recall values of clusters at each rank over the 50 queries in our test collection (Section 6.3.1) to visualize whether there is a difference between clusters at different ranks. Note that Eq. 6.1 does not take into account the size of clusters. It is intuitively undesirable if the top ranked cluster contains most relevant documents simply because most of the documents in D_q^N are assigned to it. Therefore, for each rank, we also show the box-plots for the cluster size, that is, the number of documents assigned to that cluster, normalized by the total number of results retrieved for that query.

[3] The query type information is also provided with our test collection. See Section 6.3.1.

6.3 Experimental setup

6.3.1 Test collection

As our document collection we use the Category B subset of the ClueWeb09 dataset:[4] experience with the ClueWeb09 collection suggests that the Category B subset generally contains higher quality documents than the rest of the collection [42]. It consists of 50 million English pages and is used as the test collection at the TREC 2009 Web track. As our queries, we use the TREC 2009 Web Track query set from the diversity task, which contains 50 queries, each of which comes with a set of subtopics created from query logs to reflect different facets associated with the query. While relevance judgements were made with respect to each subtopic, retrieval systems only receive a keyword query as input, i.e., short queries that usually consist of one or a few words. Moreover, each query is labeled as "ambiguous" or "faceted." In total among the 50 test queries, 12 are ambiguous and 38 are faceted.

6.3.2 Significance testing

In Section 6.4 we will conduct hypothesis testing in order to determine whether the difference between two samples are significant. These samples include inter-document similarity scores and coherence scores calculated on different document sets. None of the samples considered in this chapter is normally distributed, as tested with a Shapiro-Wilk normality test [225]. Hence, in the experiments in this chapter, we use the non-parametric Wilcoxon ranksum test [259] (MannWhitney U test [165]) for significance testing.

6.3.3 Settings for retrieval

To generate the retrieved list D_q^n, we use the Markov Random Field (MRF) retrieval model [174]; we use the full dependency model implemented by the Indri search engine[5] with default parameter settings. All follow-up clustering and experiments that involve an initially retrieved list of documents use the results generated by the MRF model. We do not apply any spam filtering on top of the baseline model.

6.3.4 Query specific clustering

We consider two types of clustering method: Latent Dirichlet Allocation [18] (LDA) and Hierarchical Clustering [128] (HC).

[4] `http://boston.lti.cs.cmu.edu/Data/clueweb09/`
[5] `http://www.lemurproject.org/indri/`

LDA We perform clustering with LDA as follows. First, we train the topic models over D_q^n with a pre-fixed number of K clusters (or latent topics). We then assign each document to a single cluster based on the topic distribution given a document. In other words, a document d is assigned to a cluster c^* such that

$$c^* = \arg\max_c p(c|d), \tag{6.2}$$

where $p(c|d)$ is estimated using the LDA model.

Hierarchical clustering Hierarchical clustering is different from LDA in nature: it is non-probabilistic and uses a vector-space representation for terms and documents. Potentially, these theoretical differences will lead to a different clustering structure.

We conduct hierarchical clustering as follows. For a query q, we create a set of clusters C, on D_q^n with Hierarchical Agglomerative Clustering (HAC).[6] For simplicity, we use cosine similarity to measure similarity between documents and use term TFIDF for document representation. We consider different linkage types, including single-linkage, complete linkage and group average (or unweighted pair group method with arithmetic mean (UPGMA)) [231].

Parameter settings for query-specific clustering For our experiments, we use the 50 test queries from the TREC 2009 Web Track and the D_q^n are the documents returned by the MRF model per query, where $n = 1000$. The number of clusters, K, is set to 10, 30, and 50. For training the latent topic models, following [35] we use the top 500 documents as D_q^n to estimate the LDA model parameters with Gibbs sampling [75, 80] and then infer the latent topic generation probabilities for all 1000 documents.

6.4 Results and discussion

6.4.1 Re-visiting the cluster hypothesis

Figure 6.2 shows the distribution of pairwise similarity scores among relevant documents (including D_q^R and $D_{q_s}^R$) versus the similarity scores between relevant and non-relevant documents (i.e., $D_q^{R\times NR}$). The raw counts are normalized by the number of pairwise similarity scores in each case. The similarity scores between relevant and non-relevant documents are mainly distributed around 0, with a very small portion distributed over high values, i.e., close to 1. In contrast, similarity scores among relevant documents are distributed relatively uniformly and with a relatively large amount of high similarity scores compare to the relevant-vs-non-relevant similarities.

Further, if we compare the similarity scores among relevant documents associated with multiple facets (D_q^R) to those associated with a single facet ($D_{q_s}^R$), we see two similar histograms, with $D_{q_s}^R$ showing slightly higher values of similarity scores than D_q^R.

[6] See Appendix A for details of HAC algorithms.

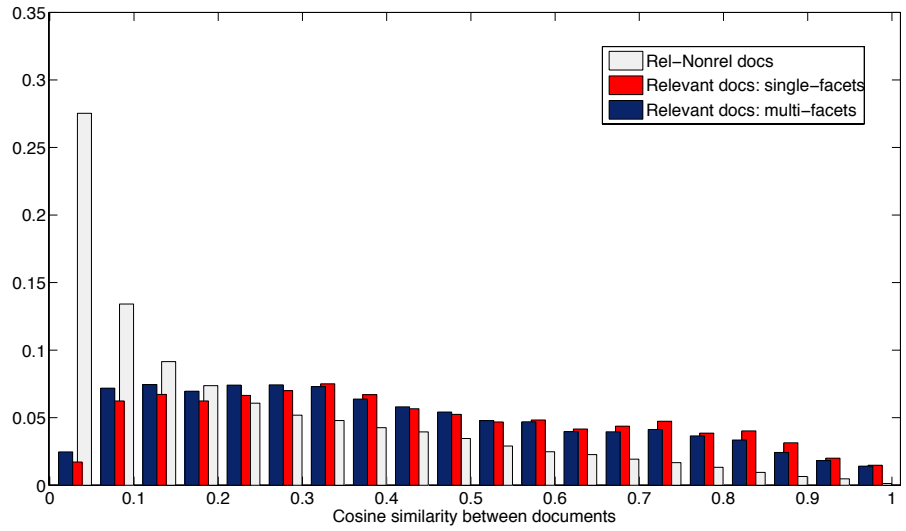


Figure 6.2: Distribution of cosine similarity scores: pairwise similarity between relevant documents vs. similarity between relevant and non-relevant documents.

Although visually similar, the difference between the two sets of similarity scores is statistically significant (p-value < 0.001.) That is, the similarity between documents associated with a single facet is higher than the similarity between documents associated multiple facets of a query.

Figure 6.3 shows a comparison of coherence scores from D_q^R, $D_{q_s}^R$ $D_q^{R\times NR}$ and D_q^{R+NR} in a box-plot. Clearly, D_q^R and $D_{q_s}^R$ show a higher coherence score than $D_q^{R\times NR}$ and D_q^{R+NR}. A significance test also confirms the same claim (p-value < 0.001). Further, the coherence scores of $D_{q_s}^R$ are significantly higher than those of D_q^R (p-value $<$ 0.001). Next, we discuss the implications given the above observations, which can be seen as an answer to our research question RQ4a.

First, the high coherence scores of relevant documents suggests that, compared to non-relevant documents, relevant documents tend to be more similar to each other. This claim holds both in the case when relevant documents are associated with multiple facets of a query as well as when relevant documents are associated with a single facet of a query.

Second, as illustrated in Section 3.3 on page 40, document sets with fewer clusters receive a lower coherence score compared to document sets with many clusters, the fact that D_q^R has a higher coherence score than D_q^{R+NR} suggests that if we cluster over the whole set of relevant and non-relevant documents for a query, it is likely that relevant documents are concentrated within a small subset of the clusters.

Third, since we have seen that there is a significant difference between the coherence scores of D_q^R and those of $D_{q_s}^R$, we need to further investigate whether cluster-based retrieval is still effective given that queries are associated with multiple subtopics. We

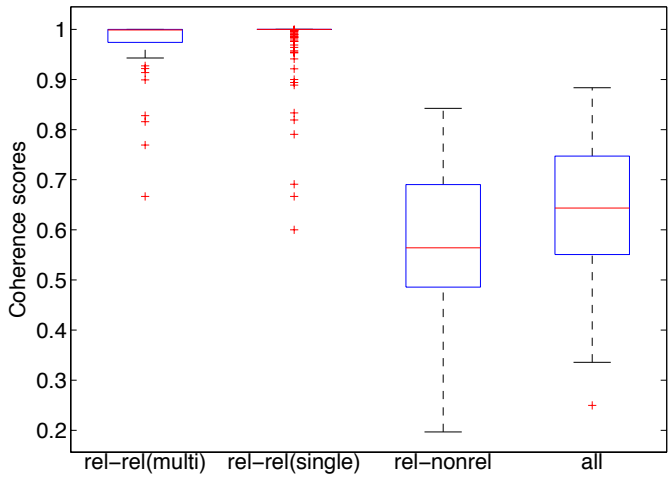


Figure 6.3: Coherence scores for 50 test queries: for judged relevant documents associated with multiple subtopics (rel-rel(multi)), for judged relevant documents associated with single subtopics (rel-rel(single)), for judged relevant and judged nonrelevant documents (rel-nonrel), and for all judged documents (all). In each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "+".

study this issue in Section 6.4.3 where we empirically seek the answer to research question RQ4c and examine whether we can generate a clustering structure desired by the cluster-based retrieval strategy for ambiguous or multi-faceted queries, such as the clustering structure described by Hearst and Pedersen [105], namely, that most relevant documents are contained in a small set of high quality clusters.

6.4.2 Ambiguous queries versus multi-faceted queries

Figure 6.4 shows the box plots of coherence scores over D_q^Rs associated with two different types of queries: ambiguous queries versus multi-faceted queries.

In Figure 6.4, the distribution of coherence scores of ambiguous queries shows a certain difference from that of the faceted queries, for example, a lower 75th percentile boundary. However, the difference between the two set of coherence scores is not statistically significant (p-value=0.81). That is, ambiguous and multi-faceted queries do not show significantly different patterns in terms of inter-document similarities.

However, after a closer look at the test collection, we find that based on the current data we have, the above conclusion may not be adequately supported. First, there are probably too few test queries for a proper statistical analysis (12 ambiguous queries

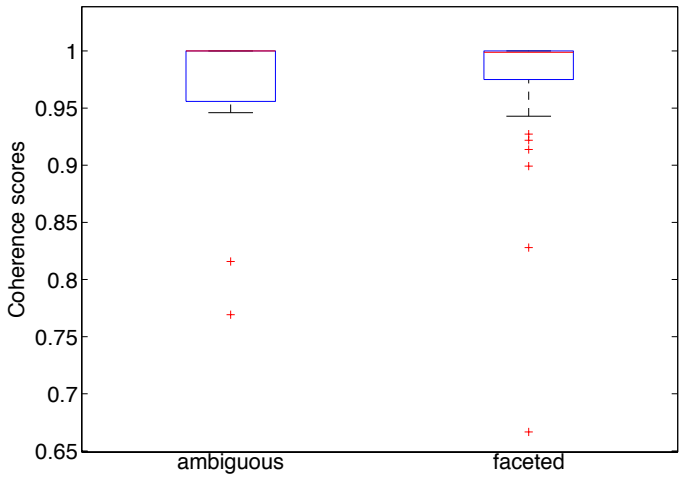


Figure 6.4: Coherence scores for 50 test queries: for judged relevant documents associated with ambiguous queries (Left), and for judged relevant documents associated with faceted queries (Right). In each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "+".

versus 38 multi-faceted queries). For some queries, the judged relevant documents only cover a single subtopic (4 out of 50), in which case the inter-document similarity among the relevant documents of these queries do not reflect the properties of a multi-faceted or ambiguous query.

Second, the categories of "ambiguous" and "multi-faceted" queries are not clearly defined. For example, query 23 "yahoo" is labeled as "ambiguous". While one may expect different interpretations of "yahoo" such as a brutish man and the Yahoo! search engine, the actual sub-topics defined for this query are as follows:[7]

1	Take me to the Yahoo! homepage.
2	Take me to Yahoo! Mail.
3	I'm looking for the Yahoo! Messenger homepage.
4	Take me to Yahoo! Finance.
5	I'm looking for the Yahoo! Music homepage.
6	I want to log in to my Yahoo! account.
7	Find information about Yahoo!, the company.

While these sub-topics are indeed "ambiguous" with respect to different sites hosted by Yahoo!, they can also be interpreted as different facets of the Yahoo! company,

[7]Full query descriptions can be found at `http://trec.nist.gov/data/web/09/wt09.topics.full.xml`.

Linkage type	K	Largest cluster			Other clusters			Uniform
		Avg.	Std.	Perc. (%)	Avg.	Std.	Perc. (%)	(%)
UPGMA	10	943.78	126.04	95.9	4.31	14.22	0.4	10.0
	30	913.66	125.17	92.6	2.37	7.41	0.3	3.3
	50	887.28	124.54	89.7	1.94	5.45	0.2	2.0
Single linkage	10	963.74	138.65	97.9	2.10	23.30	0.2	10.0
	30	943.94	137.98	95.7	1.33	12.73	0.1	3.3
	50	924.14	137.32	93.5	1.19	9.59	0.1	2.0
Complete linkage	10	356.96	118.50	36.4	69.52	43.80	7.0	10.0
	30	158.00	53.20	15.9	28.43	19.24	2.8	3.3
	50	99.58	35.17	10.0	18.02	11.86	1.8	2.0

Table 6.1: Comparison of three linkage types of the agglomerative hierarchical clustering method. *Largest cluster* shows the average size (Avg.), standard deviation (Std.) and the average percentage (Perc.) of the documents assigned to the largest cluster, calculated over the 50 test queries. *Other clusters* shows the same statistics for the rest of the clusters. *Uniform* shows the percentage of documents that should be assigned to each of the cluster if we have a uniform cluster size distribution.

depending on the granularity of the topic being defined. Since the sub-queries are constructed based on the query logs from web search engines, one can expect that the topical granularity is associated with the most popular facets or interpretations of a query. While the analysis on whether this is a proper way of constructing test queries for result diversification is outside the scope of our work, we believe that those test queries to a large extent reflect the practice of users' information needs in Web search. On the other hand, given the limitation of the test collection, we feel that there is not adequate evidence for a conclusion on whether ambiguous and multi-faceted queries are different and further investigation is necessary.

In summary, with respect to research question RQ4b, the observation made from our test collection suggests that there is no significant difference between ambiguous queries and multi-faceted queries in terms of inter-document similarities. Meanwhile, we find that there is not sufficient evidence to make a conclusion on this issue based on current data and further investigation is needed.

6.4.3 Clustering structure

Preliminary analysis of the Hierarchical Clustering (HC) algorithms

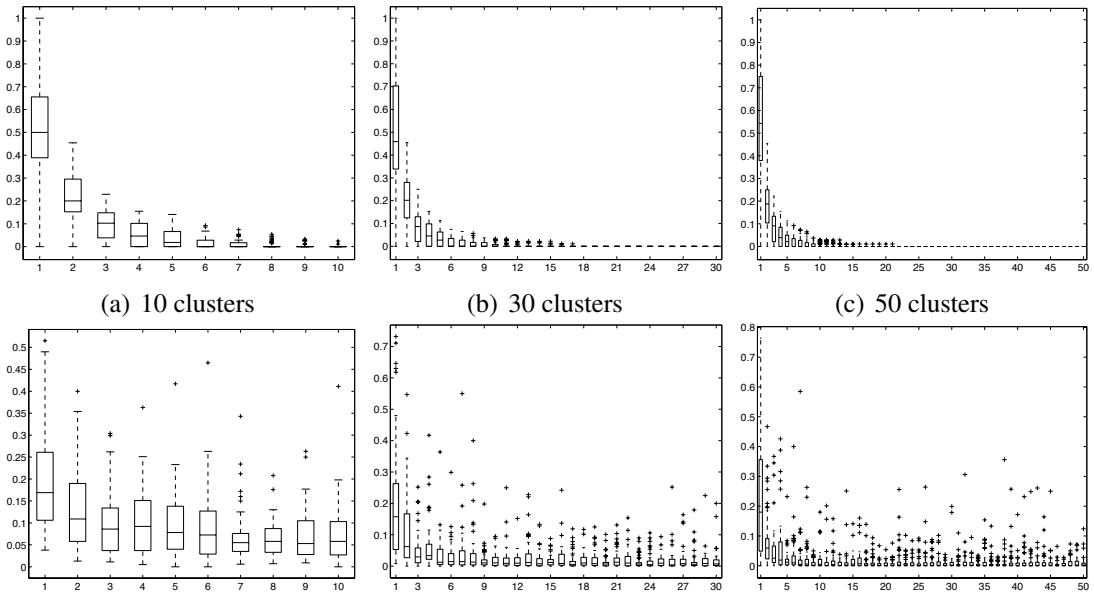
As mentioned in Section 6.1, it is undesirable if all documents are assigned to a few dominant clusters. Some of the agglomerative hierarchical clustering algorithms such as single linkage tend to generate dominant clusters, known as the "chaining effect" [167]. Here, before we proceed to empirically explore the answers to our research questions, we first conduct a preliminary experiment in order to analyze whether the hierarchical clustering algorithms we use generates a reasonable clustering structure.

In Table 6.1 we describe the properties of the clusters produced by agglomerative hierarchical clustering using three types of linkage. We see that on average, the largest cluster generated by single linkage and UPGMA constantly takes up over 90% of the documents, and each of the rest of the clusters has less than 1% of the documents, which is far from a uniform distribution of the cluster sizes. On the other hand, complete linkage generates clusters whose sizes are relatively equal, compared to the other two linkage types. For example, when $K = 10$, if the size of clusters is uniformly distributed, each cluster should contain approximately 10% of the documents. As we see in Table 6.1 in the case of hierarchical clustering with complete linkage, the average percentage of documents assigned to each of the rest of the clusters is 7.0%, which is much closer to the uniform size distribution than that of single linkage and UPGMA (0.2% and 0.4% respectively). Similar observations can be made for $K = 30$ and 50 as well. Also, the largest clusters are not as dominant as those generated by single linkage and UPGMA. Based on the above observation, we decide to continue our experiments with clusters generated with complete linkage and drop those generated by single linkage and UPGMA.

Clustering structure by LDA and HC with complete linkage

Figure 6.5 shows the distribution of relevant documents among clusters along with the cluster size distribution, where the clusters are modeled with LDA. On the one hand, we see that most relevant documents are contained in the top-ranked clusters. On the other hand, the distribution of the sizes of clusters shows a similar trend, where the top-ranked clusters tend to contain more documents than other clusters. However, this trend is more obvious in the distribution of relevant documents than that of the cluster size distribution. A more insightful observation can be made when we take a careful look at the Y-axis of the plots: on average, less than 20% of the documents are assigned to the top-ranked clusters, which, however, contain more than 50% of the relevant documents. Based on this observation, we conclude that the clusters generated by LDA have the following property: most relevant documents are contained in a small number of clusters, and this is *not* achieved by simply assigning most of the documents to those clusters.

We proceed to analyze the distribution of relevant documents in the clusters produced by hierarchical clustering with complete linkage. Figure 6.6 shows the results. We see that for clusters generated by hierarchical clustering, relevant documents are *not* equally distributed over clusters and there is a visible difference between top ranked clusters and other clusters with respect to the number of relevant documents they contain. However, the difference is not as large as that of LDA, where on average the top ranked clusters contain around 50% of the relevant documents. Here, on average, the top ranked clusters contain less than 20% of the relevant documents when $K = 10$, and less than 10% when $K = 30$ and 50. Given this observation, we can hardly claim that hierarchical clustering has generated the clustering structure required with respect to RQ4c, namely, most relevant documents are contained in a small set of high quality



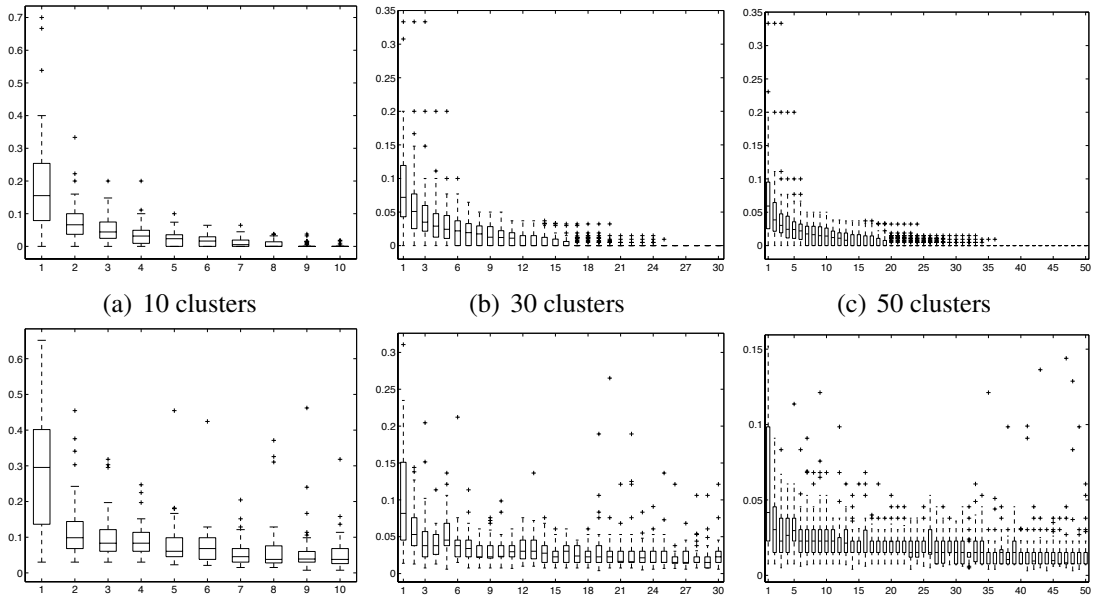
(a) 10 clusters (b) 30 clusters (c) 50 clusters

(d) 10 clusters (e) 30 clusters (f) 50 clusters

Figure 6.5: Clustering results with LDA. Figures 6.5(a)–6.5(c) show the distribution of relevant documents among clusters, over 50 test queries. Y-axis: the fraction of relevant documents contained in a cluster. Figures 6.5(d)–6.5(f) show the distribution of the size of clusters, clusters are ranked in the same order as in Figures 6.5(a)–6.5(c). Y-axis: percentage of documents assigned to a cluster. In each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "+".

clusters. Now let us have a look at the cluster size distribution. On average, the percentage of relevant documents assigned to the top-ranked cluster is almost the same as, or less than, the percentage of total documents assigned to it. For example, in the case of 10 clusters, on average, about 30% of the documents are assigned to the top-ranked clusters, which contain less than 20% of the relevant documents. The above observation suggests that in the case of hierarchical clustering with complete linkage, even if we assume that most relevant documents are contained by a few top ranked clusters, it may be due to the fact that these clusters simply contain more documents.

Further, we have learnt from the literature in cluster-based retrieval that placing high quality clusters at the top of a ranked list can effectively improve the early precision of the retrieval results. Here we re-examine this statement with our ambiguous/multi-faceted queries. Specifically, we re-rank the initial ranked list of documents D_q^N, that is, the retrieval results generated by the MRF model, by ranking the clusters. In other words, we rank the clusters in descending order of the percentage of relevant documents they contain, and keep the documents within each cluster in the original order of their retrieval scores. We then evaluate the new ranked lists of documents with Pre-



(a) 10 clusters (b) 30 clusters (c) 50 clusters

(d) 10 clusters (e) 30 clusters (f) 50 clusters

Figure 6.6: Clustering results with agglomerative hierarchical clustering with complete linkage. Figures 6.6(a)–6.6(c) show the distribution of relevant documents among clusters, over 50 queries. Y-axis: the percentage of relevant documents contained in a cluster. Figures 6.6(d)–6.6(f) show the distribution of the size of clusters, clusters are ranked in the same order as in Figures 6.6(a)–6.6(c). Y-axis: fraction of documents assigned to a cluster. In each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "+".

cision@5 and Precision@10 and check the performance of the two clustering methods in improving early precision as in an ad-hoc retrieval setting.[8]

Table 6.2 shows the results of the cluster-based document re-ranking as well as the original ranked list, that is, the MRF run. Both clustering algorithms improve over the baseline in terms of early precision. However, LDA results in a more dramatic improvement than hierarchical clustering, which indicates that the clusters generated by LDA are more effective in gathering relevant documents.

Based on the above experimental results, we arrive at the following answers to research question RQ4c. For ambiguous or multi-faceted queries, we *can* cluster the documents retrieved in response to a query in such way that most relevant documents are contained in a small set of high quality clusters. In an ad-hoc retrieval setting, we can effectively improve the early precision results for those queries by placing these

[8]We use the Qrels from TREC2009 Web track diversity task as ground truth, but discard the judgements toward different facets, i.e., if a document is relevant to one of the facets of a query, it is counted as relevant.

Method	# Clusters	Precision@5	Precision@10
MRF	–	0.1800	0.2260
HC	10	0.2480	0.2740
	30	0.3040	0.3060
	50	0.3840	0.3740
LDA	10	**0.4120**	**0.4160**
	30	**0.4720**	**0.4480**
	50	**0.4480**	**0.4300**

Table 6.2: Retrieval results with ranking clusters in terms of P@5 and P@10. MRF: baseline run; HC: ranking clusters obtained using hierarchical clustering; LDA: ranking clusters obtained with LDA.

high quality clusters at the top of a result list. However, whether this type of clustering structure can be effectively formed as well as the amount of improvement that can be achieved in terms of early precision depends on the specific clustering algorithm we use. This conclusion confirms that for ambiguous or multi-faceted queries, cluster-based retrieval strategies can be applied to improve precision oriented evaluated metrics, which provides the basis for our next chapter. In the next chapter, we will explore the effectiveness of using cluster-based retrieval in the context of result diversification.

6.5 Conclusion

In this chapter, we re-visited the Cluster Hypothesis in the context of result diversification. The main research question we considered here is

RQ4. *How do we interpret the cluster hypothesis in the context of result diversification?*

Specifically, we examined whether the hypothesis holds with respect to ambiguous or multi-faceted queries. Further, we checked whether ambiguous queries are different from multi-faceted queries in terms of inter-document similarity distributions. We also examined whether we can effectively generate clustering structure for ambiguous and multi-faceted queries such that relevant documents can be gathered in a small set of clusters, which is the basis for a cluster-based retrieval strategy in an ad-hoc retrieval setting.

Experimental results on the TREC2009 Web Track Diversity test collection shows that compared to specific or single facet queries, ambiguous/multi-faceted queries display a less coherent clustering structure, that is, they tend to contain multiple subclusters. Such a difference, although statistically significant, does not invalidate the Cluster Hypothesis. The statement "relevant documents tend to be more similar to each other than to non-relevant documents" is supported by our experimental results.

Further, we do not see significant differences between ambiguous and multi-faceted queries in terms of inter-document similarities from empirical results. Meanwhile, we also found that it is not adequate to make a conclusion based solely on observations made on the current test collection. Given the limited data at hand, we decided to leave this issue for future investigation. On top of that, we found that we can generate a clustering structure desired by the cluster-based retrieval strategy for those ambiguous/multi-faceted queries. We find that LDA based clustering is effective in clustering relevant documents into a small set of high quality clusters, without dominant clusters that contain most of the documents, while HC based clustering is not as effective.

Given the above conclusion of this chapter, we have confirmed that the cluster-based retrieval strategy can be effectively used for ambiguous or multi-faceted queries to improve precision oriented evaluation metrics. In the next chapter, we explore the use of query-specific clustering for result diversification. Particularly, we will focus on the LDA based clustering algorithm, as our prime motivation of using query-specific clustering for result diversification is to cluster relevant documents "away" from non-relevant documents, and restrict diversification to documents that are potentially relevant. In addition, we will further investigate the type of clustering structure desired by result diversification, which is intuitively different from that desired in an ad-hoc retrieval setting.

Chapter 7

Result Diversification with Query-Specific Clustering

In the previous chapter, we have re-visited the cluster hypothesis in the context of result diversification. In short, we have seen that in an adhoc retrieval setting, although the queries we consider are ambiguous or multi-faceted, the cluster hypothesis is valid and that cluster-based retrieval can effectively improve early precision of those queries. Now we continue to explore the effectiveness of using cluster-based retrieval for result diversification.

7.1 Introduction

Recently, various result diversification methods have been proposed [1, 32, 35, 37, 192, 215, 270]. Traditional retrieval strategies such as those based on the Probabilistic Ranking Principle [198] typically assume that the relevance of a document is independent from the relevance of other documents in the collection. In contrast, in the context of result diversification, the notion of "relevance" usually reflects not only the relation between a document and a given query, but also the relation between the document and other documents retrieved in response to the query. Indeed, most of the proposed diversification methods simultaneously explore *query-document* and *document-document* relations and seek to balance the two in order to address both relevance and diversity in returning retrieval results. A prime example hereof is the Maximal Marginal Relevance (MMR) approach [32], which iteratively selects documents that are most similar to the query while at the same time being most dissimilar to the documents already returned. An obvious risk with this type of diversification method is that non-relevant documents may be promoted to the top of a ranked list simply because they are different from the documents presented so far. We illustrate this phenomenon using Figure 7.1. We use MMR to rank documents for the test queries in the TREC 2009 Web track test collection [40]. In Figure 7.1, we plot three things: the change of λ, the parameter in MMR that balances relevance and diversity; Precision@10 to measure relevance; diversity,

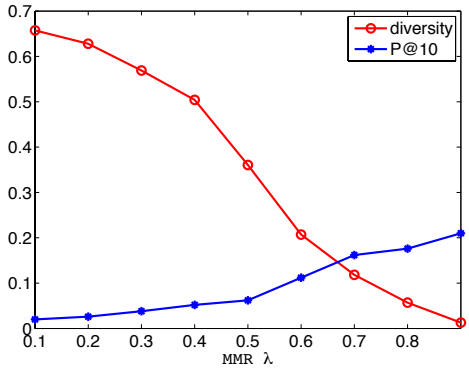


Figure 7.1: The trade-off between diversity and precision@10 for the top10 documents retrieved with MMR over different values of λ. The Y-axis shows both the precision@10 (P@10) and reversed coherence score; both scores are in the range of [0,1].

measured as one minus the Coherence Score, which we refer to as *reversed coherence score*.[1] Observe the inverse relation between diversity and relevance of the top 10 documents as we change λ. As λ increases, i.e., the emphasis on relevance is increased, there is an increase in precision but a drop in diversity, and vice versa. Ideally, a retrieval system should find the middle ground and present users with a ranked list which is both *relevant* and *diverse*.

Query-specific cluster-based retrieval is the idea of clustering retrieval results for a given query. It has long been proposed for improving retrieval effectiveness [105, 121, 137, 241]. The main intuition behind this approach to retrieval is that relevant documents tend to be clustered together. Retrieval effectiveness will be improved provided that one can place documents from high quality clusters at the top of the ranked list. Now consider a ranking approach based on query-specific cluster-based retrieval in the context of result diversification. What if we first select a set of high quality clusters (a relatively large fraction of whose documents is relevant) and then apply diversification only to the documents within these clusters? That is, what happens if we prevent documents in low quality clusters (with a limited number of relevant documents) from being promoted to the top ranks? We posit that such a strategy should lead to improved results as measured in terms of relevance and diversity because it only diversifies relevant documents. Specifically, we focus on the following question in this chapter:

RQ5 *Can query-specific clustering be used to improve the effectiveness of result diversification?*

To answer this question, we propose the following diversification framework. Given a query, we first cluster top ranked documents that are retrieved in response to the

[1]The coherence score was proposed to measure the "tightness" of a cluster of documents. Here we use one minus the coherence score of a set of documents so as to measure its "looseness", i.e., diversity. For more details see Section 7.7.1

query. We subsequently rank the clusters according to their estimated relevance to the query and apply a diversification method to the documents belonging to the top ranked clusters only. Below, we refer to this framework as *diversification with cluster ranking*.

In order to gain insight into the behavior of our proposed diversification framework, a number of specific research questions need to be addressed:

RQ5a What is the impact of the proposed diversification framework on the effectiveness of existing result diversification methods? In other words, how much performance is gained by employing query-specific clustering and applying result diversification to documents contained in the top ranked clusters only?

RQ5b What is the impact of the two main components, namely, the cluster ranker and the selection of number of top ranked clusters, on the overall performance of the proposed diversification framework?

RQ5c Further, given that we use top ranked documents retrieved in response to a query for clustering as well as for diversification, how sensitive is the performance of the proposed framework to the number of documents being selected?

RQ5d What conditions should clusters fulfill in order for diversification with cluster ranking to be effective?

We answer these research questions using empirical methods on the TREC 2009 Web track [40] test collection, as introduced in Section 6.3.1 on page 87.

The main contribution of the work presented here is two-fold. We propose a diversification framework that combines cluster-based retrieval and result diversification. The framework significantly improves the effectiveness of several result diversification methods. On top of that, we provide an in-depth analysis of the behavior of our proposed framework as well as the relation between relevance, diversity and query-specific clustering methods. Our analyses do not only help to understand the behavior of diversification with cluster ranking, but also help to direct future work on the proposed framework.

The remainder of the chapter is organized as follows. We specify the methods employed for clustering and result diversification in Section 7.2. We then introduce our proposed framework for diversification with cluster ranking in Section 7.3. We describe our experimental set-up in Section 7.4. In Section 7.5 we report on the effectiveness of diversification with cluster ranking based on our empirical results. We then proceed with two rounds of analysis. In Section 7.6, we provide a set of sensitivity analyses. We analyze the impact of the main components of our framework, that is, of methods for ranking and selecting clusters, as well as the impact of the number of documents being used for clustering and for diversification, on the overall performance of the proposed framework. Then in Section 7.7, we analyze the conditions that clusters should fulfill in order for our proposed framework to be effective. Section 7.8 concludes the chapter.

7.2 Preliminaries

We continue to use the notation as introduced in 6.2.1 on page 84. Below, we detail the clustering and diversification methods that we consider.

7.2.1 Clustering method

For clustering the documents that have been retrieved in response to a query, we use LDA as described in 6.3.4 on page 87.

We choose LDA for three main reasons. First, it models the relation between words, documents and clusters (that is, latent topics) within a theoretically sound probabilistic framework. Once the topic models have been obtained, it is convenient to infer the generation probability of a cluster for an arbitrary piece of text. In our case, we use the trained model to infer the probability of a cluster generating a query as an estimation of the relevance relation between the cluster and the query, see Section 7.3.2. Second, the latent topics can be seen as the potential "facets" of a query. Although the main purpose of applying query-specific clustering is to gather relevant documents instead of modeling query facets, the latent topic underlying a cluster addresses both. Particularly, we can apply the same LDA model for facet modeling when implementing diversification methods that explicitly model the potential facets of a query, as we will see in Section 7.2.2. Third, as discussed in 6.3.4, in the adhoc retrieval setting, the clustering structure generated by LDA collects relevant documents more effectively than the structure generated by HC. Our prime motivation for using query-specific clustering for result diversification is to exploit its ability to collect relevant documents and to promote precision, and therefore LDA provides a good starting point.

We will further investigate the type of clustering structure desired by our proposed diversification framework and discuss diversification performance using clustering algorithms other than LDA in Section 7.7.

7.2.2 Diversification methods

In our experiments we consider the following diversification methods: MMR, FM-LDA, IA-select and RR. These will be explained next.

Maximal marginal relevance (MMR) According to the MMR method [32], a document d is selected for inclusion in a ranked list of documents for a given query q such that

$$d = \arg\max_{d_i \in R} [\lambda \cdot sim_1(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in S} sim_2(d_i, d_j)], \tag{7.1}$$

where S is the set of documents that have been selected so far and R is the set of candidate documents to be selected; sim_1 is the similarity between query and document and sim_2 is the similarity between two documents. For sim_1 and sim_2 we can use any type of similarity measure; we specify our choices in Section 7.4.4.

Facet model with LDA (FM-LDA) We also consider the FM-LDA model [35], with marginal likelihood as optimization method. Given a set of documents $D = \{d_i\}_{i=1}^{n}$, the model uses LDA to capture a set of hypothesized facets $F = \{f_j\}_{j=1}^{m}$, and a subset of D is selected such that the following likelihood function is maximized:

$$L(y_i|F,D) = \prod_{j=1}^{m} \left(1 - \prod_{i=1}^{n} (1 - p(f_j \in d_i))^{y_i} \right), \tag{7.2}$$

where $y_i = 1$ if document d_i is selected and $y_i = 0$ otherwise; $p(f_j \in d_i)$ denotes the probability that facet f_j is covered by document d_i. The likelihood function is maximized subject to the constraint $\sum y_i \leq l$, where l is a predefined number of documents that are to be returned in the ranked list. In practice, a greedy approach is applied, which selects a document that maximizes the likelihood function conditioned on all the documents that have already been selected.

Note that FM-LDA identifies facets of a query with LDA, which is very similar to how our document clustering method identifies clusters, cf. Section 7.2.1. There are two distinguishing differences. First, the underlying assumptions on latent topics are different: in FM-LDA, the trained latent topics are expected to reflect the underlying facets of a query, while in document clustering, we do not care whether the latent topics can accurately reflect the actual facets of a query. Second, in document clustering, we assign each document to a single cluster, cf. Eq. 6.2, while in FM-LDA there is no need for assigning documents to latent topics. Also, as we see from Eq. 7.2, FM-LDA treats all facets as identified by LDA in the same manner. Contrary to our method, FM-LDA does not consider the importance of a facet, that is, some facets may be more relevant than others. If we assume that document clusters reflect the potential facets of a query, our method takes into account the importance of each facet via cluster ranking. We will see the impact of ranking clusters on the diversification result of FM-LDA in Section 7.5.

Intent aware select (IA-select) With IA-select [1] the selection of a document is determined by its relevance to the query as well as the probability that it satisfies potential facets given that all previously selected documents fail to do so. Given a candidate document set and a set of potential facets F, the algorithm selects the document to be included in the returned set S from a candidate set R that maximizes the *marginal utility* at each step:

$$d^* = \arg\max_{d \in R} \sum_{f \in F} P(f_i|q,S)\, V(d|q,f_i), \tag{7.3}$$

where $V(d|q,f)$ is a quality value of d that is computed using the retrieval score of d with respect to q, weighted by the likelihood that d belongs to f. Further, $P(f|q,S)$ is the conditional probability that q belongs to f, given that all documents in S failed to provide information on f:

$$P(f|q,S) = (1 - V(d|q,f))P(f|q,S \setminus \{d\}). \tag{7.4}$$

Instead of training a classifier with a taxonomy as implemented in the original IA-select algorithm to obtain $P(f|q,S)$ and the likelihood that d belongs to f, we estimate these probabilities with the topic distribution from the LDA model. Similar to FM-LDA, we use the same LDA model for clustering and facet modeling.

Round-Robin facet selection (RR) This approach naturally arises in the setting of our strategy for diversification with cluster ranking (defined in the next section). For a given set of documents D, we generate a set of K clusters with LDA, and rank the clusters according to a certain ranking criterion, for example, in descending order of the relevance of the clusters to a given query, which results in a ranked list of clusters $RC = c_1, \ldots, c_k$. For each cluster, we rank the documents within that cluster in the order of their original retrieved scores. We then select documents belonging to different clusters in a round-robin fashion. That is, in each round, we take the top ranked documents from each of the clusters and add them to the new ranked list in the order of $c_1, \ldots, c_k$. This selection procedure continues until no documents are left in any of the clusters. The motivation behind this approach is as follows. By clustering documents, we gather documents with similar content within the same cluster, while documents from different clusters contain diverse content. Intuitively, we can see the clusters as different facets associated with a given query. Hence, selecting documents from different clusters should result in a diverse result list. On top of that, by selecting the documents in the order of the ranking of the clusters, we take into account the importance of different facets.

7.3 Result diversification with cluster ranking

In this section, we introduce our proposed framework for combining query-specific clustering and result diversification. The overall goal of the approach is to *rank clusters with respect to their relevance to the query and to limit the diversification process to documents contained in the top ranked clusters only, in order to improve the effectiveness of diversification as measured in terms of both relevance and diversity.*

7.3.1 Proposed framework

Assume that we have a ranking method $cRanker(\cdot)$ that ranks clusters with respect to their relevance to a query and a diversification method $Div(\cdot)$ that diversifies a given ranked list of documents. We propose the following procedure for diversification. The input of the procedure is the output of *cRanker*, that is, a ranked set of clusters $RC = c_1, \ldots, c_K$, where $c_1 \succ c_2 \succ \ldots \succ c_K$, and the documents contained in each cluster, D_q^c. A free parameter T is used to indicate the number of top ranked clusters to be selected for diversification. Furthermore, $dRanker(\cdot)$ is assumed to be a document ranker that ranks documents according to certain criteria, for example, ranking

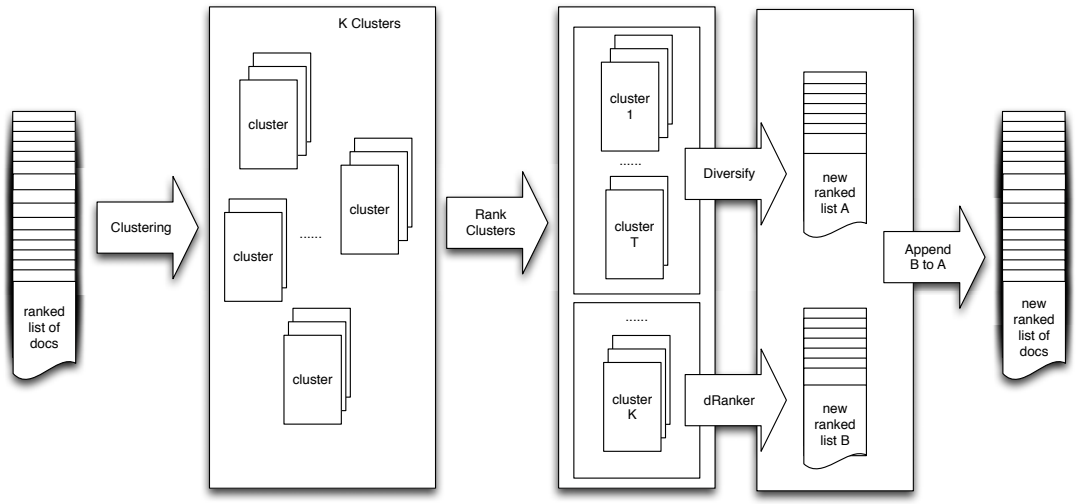


Figure 7.2: Diversification with cluster ranking. The input is a ranked list of documents and output is a diversified ranked list of documents. The arrows represent methods applied to the documents, and the boxes show the status of the documents.

Algorithm 1 Diversification with cluster ranking

Input: $Div(\cdot)$, $RC = c_1, \ldots, c_k$, $\{D_q^c\}$, T
Output: re-ranked documents *ranked*
$ranked = \emptyset$
$to\_rank = \{D_q^{c_i}\}_{i=1}^{T}$
$ranked \leftarrow ranked \cup Div(to\_rank)$
for i in $T+1$ to k **do**
  $ranked \leftarrow ranked \cup dRanker(D_q^{c_i})$
end for
return *ranked*

documents in descending order of their retrieval scores. We illustrate the proposed diversification framework in Figure 7.2.

The pseudo code of our diversification with cluster ranking method is given in Algorithm 1. It applies $Div(\cdot)$ to the documents assigned to the top T ranked clusters; documents assigned to clusters ranked below the top T are ranked by $dRanker(\cdot)$ and appended to the ranked list of documents obtained from the top T clusters.

Two crucial components of our proposed diversification framework are the function $cRanker(\cdot)$ that ranks the clusters and the selection of the cut-off value T. In the following sub-sections, we discuss our choices for these two components. As for $Div(\cdot)$, we use the diversification approaches introduced in Section 7.2.2.

7.3.2 Cluster ranking

As we pointed out above, ranking clusters based on their relevance to a query is an important issue, which has been studied in the context of cluster-based retrieval. Since our main purpose is not to develop a new method for ranking clusters, we only discuss two ways to rank clusters that are necessary for investigating the effectiveness of our proposed framework for result diversification.

Query likelihood For a query q, we rank the clusters in descending order of the probability $p(c|q)$, which is inferred from the LDA model as described in Section 7.2.1. In other words, the clusters are ranked according to their likelihood given the query. This is a simple but reasonable approach. Presumably, if a cluster has a high probability of generating a query, the documents contained in this cluster are more likely to be relevant to the query. Hence, the cluster is more likely to contain relevant documents.

Oracle ranker We also consider an oracle ranker, that is, a ranker that uses information from explicit relevance judgements. Here, the probabilities $p(c|q)$ are estimated using the judgments of retrieved documents in D_q^n. It is computed as

$$p(c|ora_q) = \frac{|D_q^c \cap D_q^R|}{|D_q^c|}. \tag{7.5}$$

In words, using $p(c|ora_q)$, we rank clusters according to the number of relevant documents contained in them, normalized by the size of the cluster.

Observe that Eq. 7.5 combines two important factors: the number of relevant documents in c_k and its relative size. As discussed in Section 6.3.4, we hope that the top ranked clusters contain most of the relevant documents, which is not achieved by simply assigning most of the documents to a single huge cluster. In Section 7.7 we will discuss the properties of the clustering structure desired by our proposed framework in more detail.

7.3.3 Determining the cut-off T

The optimal number of top-ranked clusters whose documents will be used for diversification, T, depends on a number of factors: the diversification method, the total number of clusters (that is, K), the evaluation metric, as well as the query. Similar to our strategy for ranking clusters, we discuss two ways to determine the value of T, namely, automatically determining T with cross-validation and using an oracle.

Automatically determining T using cross-validation over queries Automatically determining the optimal cut-off T is non-trivial. We typically do not have sufficiently many test queries to learn the optimal value of T, hence we apply leave-one-out cross-validation to find the optimal value of T for each query. Specifically, we optimize

T over a set of training queries for a given K and a given diversification method for a given evaluation metric by exhaustive search, i.e., over all possible values of $T = 1, ..., K$. Then we apply the learned T on the test query.

Oracle T To obtain the oracle value of T, for each query, we find the optimal T for each diversification method and each setting of K over each single evaluation metric, i.e., the performance is maximized in terms of a corresponding evaluation metric. For a given cluster ranking approach, the oracle T provides an upper bound for our proposed diversification framework under the given setting, which shows the potential merit of applying cluster ranking and selection for result diversification.

7.4 Experimental setup

In this section, we describe our experimental setup for investigating the effectiveness of diversification with cluster ranking. We begin by recalling the research questions raised in Section 7.1. Then we specify the settings for our experiments, including the evaluation metrics and the parameter settings for the retrieval method, diversification methods and clustering algorithms.

7.4.1 Research questions and experiments

The main research question we address in this chapter is:

RQ5 *Can query-specific clustering be used to improve the effectiveness of result diversification?*

More specifically, we investigate the following:

RQ5a What is the impact of diversification with cluster ranking on the effectiveness of existing result diversification methods? In other words, how much performance is gained by employing query-specific clustering and applying result diversification to documents contained in the top ranked clusters? In particular, given the query likelihood cluster ranker and an automatically determined value of T, what is the effectiveness of the proposed diversification framework?

We apply Algorithm 1 with various diversification methods, i.e., various instances of $Div(\cdot)$, on an initially retrieved ranked list of documents D_q^n where $n = 1000$. We write cX to denote the instance of Algorithm 1 where X is used as $Div(\cdot)$. First, we take the cluster ranker based on query likelihood (Section 7.3.2), and investigate whether the proposed diversification framework is effective even though the ranking of clusters may not be optimal. We set T to different values and compare the results of only diversifying over documents contained in the top T clusters to the result of diversifying over the complete ranked list of documents. Then, in order to evaluate the performance

of our framework combined with the query likelihood cluster ranker and the automatically determined T, we use cross-validation as described in Section 7.3.3 to determine the optimal T for each diversification and a given K. While optimizing T on training queries, we use two evaluation metrics: α-NDCG@10 for α-NDCG based metrics and IA-P@10 for IA-P based metrics (see Section 7.4.3 for a description of these evaluation metrics).

We analyze the effectiveness of diversification with cluster ranking along four dimensions: the cluster rankers used, the cut-off value T, the number of documents used for clustering as well as for diversification, and the clustering algorithms used. In our experiments, the query likelihood cluster ranker and the method to automatically determine T are chosen for simplicity, while many other possibilities exist. Insights into the roles of both components and their interactions within our proposed framework are useful for future work on potentially more effective approaches to ranking clusters and to automatically determining T. In addition, the number of documents being included from the initially retrieved ranked list for clustering and for diversification can be seen as an additional free parameter. We provide a comprehensive analysis of the sensitivity of the proposed framework to the choice of this parameter. Also, LDA is used for clustering for the reasons stated in Section 7.2.1. It is useful to examine the general properties of the sort of clustering structure desired by Algorithm 1, as this provides guidance for choosing suitable clustering algorithms. Specifically, then, we seek to answer the following additional research questions:

RQ5b What is the impact of the two main components, namely, the cluster ranker and the selection of the number of top ranked clusters, on the overall performance of diversification with cluster ranking?

RQ5c Further, given that we use top ranked documents retrieved in response to a query for clustering as well as for diversification, how sensitive is the performance of the proposed framework to the number of documents being selected?

RQ5d What conditions should clusters fulfill in order for diversification with cluster ranking to be effective?

In order to answer RQ5b, we conduct a set of "oracle" runs in three settings. First, we analyze the impact of T by comparing the diversification results using the oracle T and the predicted T determined by cross-validation. Then, we analyze the impact of the cluster ranker by comparing the diversification performance using the oracle $cRanker(\cdot)$ as described in Section 7.3.2 to that of the query likelihood-based cluster ranker. In addition, we combine the oracle cluster ranker and the oracle T so as to identify an upper bound on the improvement of diversification with cluster ranking over diversification without cluster ranking and selection; see Section 7.6.1.

In order to answer RQ5c, we continue using the oracle cluster ranker and conduct a set of three experiments with varying number of documents for clustering and for diversification. Given an initially retrieved ranked list of documents, let us refer to the documents used for clustering, i.e., training the LDA model, as D_q^C and the documents to which we apply Algorithm 1 as D_q^D. The settings of the three experiments can be described as follows.

Setting 1. Set $C = 100, 300, 500$ and $D = 1000$. In this setting, we fix the number of documents used for applying Algorithm 1 and compare the impact of LDA models trained on different number of top ranked documents on our proposed diversification framework.

Setting 2. Set $C = 500$ and $D = 100, 300, 1000$. In this setting, we fix the number of documents used for training the LDA model and analyze the effect of applying Algorithm 1 on different number of top ranked documents.

Setting 3. Set $C = 100, 300, 500$ and $D = 100, 300, 1000$, respectively. In this setting, we check the performance of our proposed framework by varying the number of documents for both clustering and for diversification simultaneously.

Note that in each setting, when $C = 500$ and $D = 1000$, it is the default parameter setting of the experiments discussed above (see Section 7.4.4) and we use the results of this parameter setting as baselines in our analysis. See Section 7.6.2 for details.

In order to answer RQ5d, we hypothesize conditions that should be fulfilled by the clustering structure generated by a clustering algorithm based on literature on cluster-based retrieval as well as the characteristics of the diversification task. On top of that, we include hierarchical clustering as an alternative clustering algorithm that generates a clustering structure different from that generated by LDA. We examine the impact of the conditions on clustering structure by comparing the properties of the two types of clustering structure and the end performance of our diversification with cluster ranking framework. See Section 7.7 for details.

7.4.2 Test collection

We use the TREC 2009 Web track [40] test collection, as introduced in Section 6.3.1 on page 87.

7.4.3 Evaluation metrics

For evaluation, we use α-NDCG [41], which adapts the NDCG measure to address both relevance and diversity. The parameter α denotes the probability that a user is still interested in a document given that the facet associated with the document is already covered by previously seen documents. By default, we set α to 0.5. Also, we use the

IA-P measure [40] with a uniform distribution for judged facets. See Section 2.4.2 on page 29 for a detailed description of the two measures.

A paired t-test is used for testing the significance of the difference between run results as indicated in the captions: $^{\triangle}$ ($^{\triangledown}$) indicates that an improvement (decline) is significant with $\alpha < 0.05$; $^{\blacktriangle}$ ($^{\blacktriangledown}$) indicates that an improvement (decline) is significant with $\alpha < 0.01$.

7.4.4 Parameter settings

Settings for retrieval For our baseline retrieval method, we use the MRF retrieval model with the settings described in Section 6.3.3 on page 87.

Settings for clustering For clustering, we use the same setting as described in Section 6.3.4 on page 88. We use LDA for exploring answers to research questions RQ5a, RQ5b and RQ5c. In Section 7.7, we will consider both LDA and HC with complete linkage in order to seek answers to RQ5d.

Settings for diversification The diversification methods that we consider come with the following model parameters:

MMR. For sim_1 we normalize retrieval scores into $[0,1]$ (see below); for sim_2, we use cosine similarity. To determine λ, we performed a simple parameter sweep by applying MMR without cluster ranking and use α-NDCG@10 as the optimization metric, that is, we chose the λ that generates the best result in terms of α-NDCG@10; λ was found to be 0.9. Optimization is performed with respect to diversification with entire ranked list.

IA-select. We model the distribution of facets of a query with the cluster distribution inferred by LDA (see Section 7.2.2). Specifically, the importance of a cluster, that is, facet, for a query q is determined by $p(c|q)$, which is inferred from the trained LDA model.

FM-LDA. Similar to IA-select, facets of a query are discovered by LDA; the only parameter is the number of facets.

RR. We order clusters by descending value of $p(c|q)$ which is inferred in the same way as for IA-select.

Score normalization For MMR and IA-select, the original retrieval scores are involved for diversification. In our experiments, we normalize those scores into the range [0, 1] in order to combine scores with different ranges. Since the original retrieval score is usually in the log domain, we first transform it back to its original domain, and then for the score of each document s_d in the ranked list D_q^n, we normalize it using $norm(s_d) = s_d / \sum_{i \in D_q^n} s_i$.

7.5 Experimental results

In this section, we discuss the results of experiments that aim to answer the main research question: *Can query-specific clustering be used to improve the effectiveness of result diversification using diversification with cluster ranking?*

7.5.1 Effectiveness of diversification with cluster ranking

How does diversification with cluster ranking compare to diversification over the complete ranked list of documents? Figure 7.3 shows the trends of the performance of each diversification method with cluster ranking (cMMR, cFM-LDA, cIA-select and cRR) across values of T, the number of top-ranked clusters whose documents are used for diversification. For each method, when $T = K$, diversification with cluster ranking is equivalent to diversifying the complete list of initially retrieved documents. Here, we only show the results measured using α-NDCG@10 and IA-P@10 for $K = 10$, 30 and 50; a similar trend can be observed for α-NDCG@X and IA-P@X, for $X = 5, 20$.

For all methods, the plots in Figure 7.3 show that diversification does not benefit from, or is even hurt by, selecting all clusters, that is, by diversifying the complete ranked list of documents. Also, for each method there is an optimal value of T that maximizes the performance of the method, which is smaller than the total number of clusters, that is, for which the optimal value of T satisfies $T < K$. If we could accurately find this optimal T, the diversification performance is bound to be more effective than diversification over the complete ranked list of documents. We conclude from this observation that, for a given cluster ranker, the proposed framework has the potential to improve the diversification effectiveness if a proper T is chosen. In the following sections, we will further examine whether the difference between diversification with entire ranked list and diversification with selected T clusters is significant, where the selected T can be determined through cross-validation as well as set by oracle.

We therefore investigate the effectiveness of diversification with cluster ranking based on the query-likelihood cluster ranker combined with the predicted T next.

7.5.2 Diversification with the query likelihood-based cluster ranker and predicted T

Now let us look at the performance of diversification using query likelihood for ranking clusters and using cross-validation to predict the number T of top-ranked clusters to be considered for diversification. Tables 7.1–7.4 compare diversification with cluster ranking against diversifying the complete list of retrieved documents. As before, cX indicates the runs with cluster ranking and selection, where X is the name of a diversification method; in each table, K is the total number of clusters. We also list the average predicted value of T. On top of that, we include the performance achieved by each method when T is optimal, which is indicated by T^*, e.g., the peak points in Figure 7.3. Note that T^* is different from the oracle T: in the case of oracle T, the value

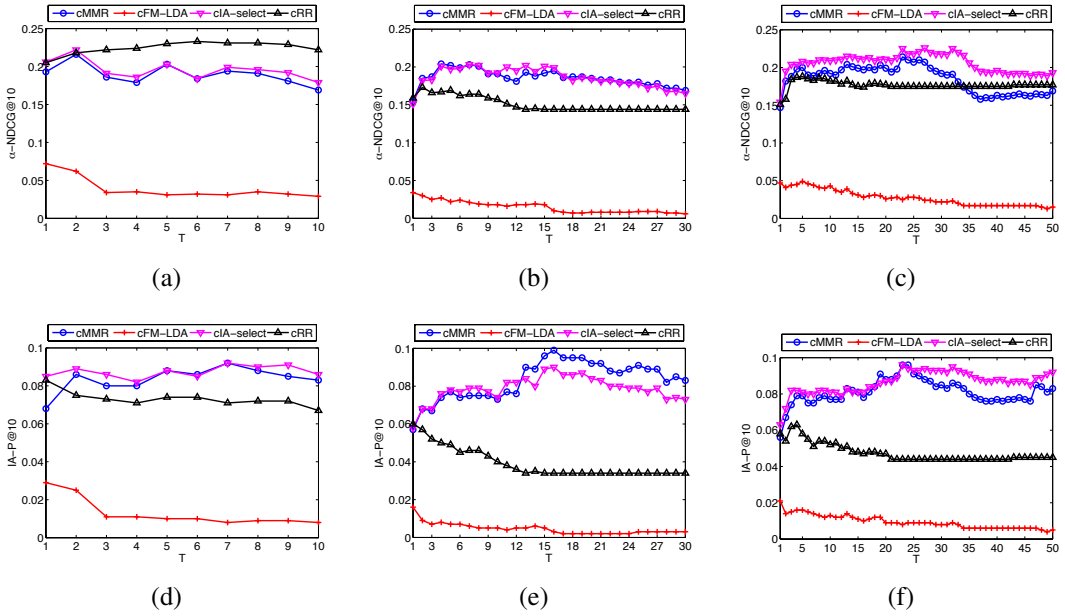


Figure 7.3: Diversification with cluster ranking using query likelihood as $cRanker(\cdot)$ over different numbers of selected top ranked clusters (T). The evaluation metrics are α-NDCG@10 (top row) and IA-P@10 (bottom row). The total number of clusters K is set to 10 (7.3(a) and 7.3(d)), 30 (7.3(b) and 7.3(e)) and 50 (7.3(c) and 7.3(f)). Note that the plots have different scales on the Y-axis for different evaluation metrics.

of T is optimized for each query, while T^* is optimized for the average performance over all queries.

We see that for different diversification methods, diversification with cluster ranking outperforms the original algorithms in nearly all cases, even though query likelihood is not a perfect ranker for ranking clusters and T has not been fully optimized. If we take the optimal T with respect to the average performance over all queries, i.e., T^*, we see further improvements, and more improvements are statistically significant compared to that of the predicted T. In some cases, the average predicted T is very close to the T^* and result in similar performance. However, a small difference between the average predicted T and T^* does not necessarily lead to a small difference between the diversification results. This may be because the difference between the average predicted T and the T^* does not reflect the per-query difference, which can in fact lead to very different results.

Below, we take a close look at the performance of individual diversification methods, focusing on the results obtained using automatically determined T. Results obtained by T^* are listed for completeness, but not discussed further.

For MMR (Table 7.1), we see that in all cases except when $K = 10$ and for IA-P@10, the performance of diversification with cluster ranking improves over the original diversification algorithm, although the improvements are not always statistically

K	Method	α-NDCG@5		α-NDCG@10		IA-P@5		IA-P@10	
		score	avg. T	score	avg. T	score	avg. T	score	avg. T
–	MMR	0.122	–	0.169	–	0.066	–	0.083	–
10	cMMR	**0.191**$^{\triangle}$	1.98	**0.216**	2.00	0.070	2.44	0.069	6.82
	cMMR$^{T^*}$	**0.191**$^{\triangle}$	2	**0.216**	2	**0.090**	2	**0.092**	7
30	cMMR	0.157	4.42	0.171	4.76	0.077	13.54	0.090	15.94
	cMMR$^{T^*}$	**0.179**$^{\triangle}$	4	**0.204**	4	**0.085**	16	**0.099**	16
50	cMMR	0.178$^{\triangle}$	21.80	**0.214**$^{\triangle}$	23.00	0.090	23.00	**0.096**	23.00
	cMMR$^{T^*}$	**0.179**$^{\triangle}$	23	**0.214**$^{\triangle}$	23	**0.092**	23	**0.096**	23

Table 7.1: Results of MMR vs. cMMR. For each K and each evaluation metric, the performance of cMMR is compared to the corresponding performance of MMR. Boldface indicates the best score achieved for a given K. For cMMR$^{T^*}$, the avg. T is the value of T^*.

significant.

K	Method	α-NDCG@5		α-NDCG@10		IA-P@5		IA-P@10	
		score	avg. T	score	avg. T	score	avg. T	score	avg. T
10	FM-LDA	0.027	–	0.029	–	0.011	–	0.008	–
	cFM-LDA	**0.058**	1.00	**0.072**$^{\triangle}$	1.00	**0.031**$^{\triangle}$	1.00	**0.029**$^{\triangle}$	1.00
	cFM-LDA$^{T^*}$	**0.058**	1	**0.072**$^{\triangle}$	1	**0.031**$^{\triangle}$	1	**0.029**$^{\triangle}$	1
30	FM-LDA	0.000	–	0.006	–	0.000	–	0.003	–
	cFM-LDA	0.020$^{\triangle}$	2.06	0.027$^{\triangle}$	1.02	0.009$^{\triangle}$	1.00	**0.016**$^{\triangle}$	1.96
	cFM-LDA$^{T^*}$	**0.022**$^{\triangle}$	2	**0.034**$^{\triangle}$	1	**0.010**$^{\triangle}$	2	**0.016**$^{\triangle}$	1
50	FM-LDA	0.008	–	0.015	–	0.004	–	0.005	–
	cFM-LDA	0.020	1.32	0.026	4.60	**0.021**$^{\triangle}$	1.00	**0.021**$^{\blacktriangle}$	1.00
	cFM-LDA$^{T^*}$	**0.038**$^{\blacktriangle}$	1	**0.049**$^{\triangle}$	5	**0.021**$^{\blacktriangle}$	1	**0.021**$^{\blacktriangle}$	1

Table 7.2: Results of FM-LDA vs. cFM-LDA. For each K, the results of cFM-LDA are compared to the corresponding results of FM-LDA. Boldface indicates the best score achieved for a given K. For cFM-LDA$^{T^*}$, the avg. T is the value of T^*.

For FM-LDA (Table 7.2) we see that in all cases, diversification with cluster ranking improves over diversification without cluster ranking; in most cases the improvement is statistically significant. Also, we notice that the average number of selected top ranked clusters in each case is small compared to other methods (that is, cIA-select, cMMR and cRR). In other words, when more clusters are included for diversification, the performance of FM-LDA drops quickly. This phenomenon suggests that FM-LDA may be very sensitive to non-relevant documents: including more clusters increases the chance of including more non-relevant documents for diversification and the performance of FM-LDA decreases in this situation.

For IA-select (Table 7.3), we see that in most cases, performance is improved by applying diversification with cluster ranking. Exceptions include the following cases: $K = 10$ using IA-P@5 and IA-P@10, $K = 30$ using α-NDCG@10 and $K = 50$ using IA-P@10 where the performance stays the same.

For RR (Table 7.4), in all cases except when $K = 50$ using α-NDCG@10, diversifi-

K	Method	α-NDCG@5		α-NDCG@10		IA-P@5		IA-P@10	
		score	avg. T	score	avg. T	score	avg. T	score	avg. T
10	IA-select	0.125	–	0.179	–	0.069	–	0.086	–
	cIA-select	**0.199**$^{\triangle}$	2.00	0.221	2.00	0.053	3.30	0.056	6.58
	cIA-select$^{T^*}$	**0.199**$^{\triangle}$	2	**0.222**	2	**0.096**	7	**0.092**	7
30	IA-select	0.116	–	0.165	–	0.063	–	0.073	–
	cIA-select	0.145	7.00	0.158	7.64	0.079	14.36	0.077	16.00
	cIA-select$^{T^*}$	**0.185**$^{\triangle}$	7	**0.203**	7	**0.094**$^{\triangle}$	16	**0.090**$^{\triangle}$	16
50	IA-select	0.146	–	0.193	–	0.078	–	0.092	–
	cIA-select	0.181$^{\triangle}$	15.06	0.208	27.14	0.100	31.36	0.092	23.54
	cIA-select$^{T^*}$	**0.199**$^{\triangle}$	9	**0.226**$^{\triangle}$	27	**0.105**$^{\triangle}$	32	**0.096**	23

Table 7.3: Results of IA-select vs. cIA-select. For each K, the results of cIA-select are compared to the corresponding results of IA-select. Boldface indicates the best score achieved for a given K. For cIA-select$^{T^*}$, the avg. T is the value of T^*.

K	Method	α-NDCG@5		α-NDCG@10		IA-P@5		IA-P@10	
		score	avg. T	score	avg. T	score	avg. T	score	avg. T
10	RR	0.198	–	0.222	–	0.079	–	0.067	–
	cRR	0.199	2.68	**0.233**$^{\triangle}$	6.00	0.085	2.00	**0.083**	1.00
	cRR$^{T^*}$	**0.204**	2	**0.233**$^{\triangle}$	6	**0.091**	2	**0.083**	1
30	RR	0.137	–	0.144	–	0.049	–	0.034	–
	cRR	0.151	2.94	0.168	2.06	0.065	2.90	**0.060**$^{\triangle}$	1.00
	cRR$^{T^*}$	**0.152**	2	**0.173**$^{\triangle}$	2	**0.068** $^{\triangle}$	2	**0.060**$^{\triangle}$	1
50	RR	0.157	–	0.177	–	0.057	–	0.045	–
	cRR	0.160	5.00	0.172	4.86	0.067	3.20	0.056	3.92
	cRR$^{T^*}$	**0.176**$^{\triangle}$	5	**0.188**	5	**0.072**	3	**0.063**$^{\triangle}$	4

Table 7.4: Results of RR vs. cRR. For each K, the results of cRR are compared to the corresponding results of RR. Boldface indicates the best score achieved for a given K. For cRR$^{T^*}$, the avg. T is the value of T^*.

cation with cluster ranking outperforms the original method. Note that ranking clusters is inherent for RR and the only difference between RR and cRR is that cRR applies RR to the top T selected clusters. The improvement of cRR over RR shows that eliminating from the diversification process clusters that are likely to be non-relevant to the query can effectively improve the result diversification performance.

Finally, we take a look at cases where diversification with cluster ranking does not outperform its original counterparts. Let us use cIA-select as an example. If we look at the corresponding plots in Figure 7.3(b), 7.3(d) and 7.3(f) for the cases where cIA-select loses against IA-select, we see that the performance curves of cIA-select across different cut-off values T fluctuate frequently and on each curve, several local maximums exist and the differences between those local maximums are small. On the one hand this may create difficulties for the cross-validation approach to find a global optimal T; on the other hand, this indicates that the ranking of clusters needs to be improved. Similar observations can be made for cMMR and cRR.

7.5.3 Additional remarks

Although not directly related to our experimental objectives, in Table 7.5 we show the performance of the initial retrieval result generated by the MRF model, as measured using diversification metrics. We compare its performance to that of applying diversification methods, and of the results of diversification with cluster ranking. For the results of diversification with cluster ranking, we only show the runs with best performance among different K values, in terms of α-NDCG@10 and IA-P@10. Note that the value of K that results in the best performance may differ for different diversification methods, which suggests that for optimizing performance and a careful model selection, K should be tuned separately for each diversification method and metric.

Diversification with cluster ranking outperforms diversification over the complete ranked list of documents, but does not always outperform the baseline, that is, the initial ranked list returned by MRF. The performance of diversification with cluster ranking is closely related to the performance of the underlying diversification methods: diversification methods that perform better, e.g., IA-select and RR, result in better performance with cluster ranking.[2] The performance of FM-LDA is low in general, which may be due to the fact that it retrieves too few relevant documents after diversification, as was also found by Carterette and Chandar [35].

In Table 7.5 We notice that RR and its cluster-based version cRR, while simple, are very effective compared to other diversification methods. The effectiveness of RR and cRR may be due to the following. By applying RR, we first need to rank the clusters, which potentially improves the early precision. On top of that, we select documents from different clusters in a round-robin fashion, which promotes diversity. On top of that, cRR cuts the clusters at top T, which further prevents potentially non-relevant clusters from being included for diversification.

7.5.4 Answers to the main research question

We turn to our main research question RQ5a, for which we have obtained the following answers. First, with an imperfect cluster ranker, diversification using documents from a carefully selected number of top-ranked clusters can be more effective than diversification using all documents in the initial retrieved list. Second, in general, the query likelihood-based cluster ranker and the predicted T are effective for improving the performance of the diversification methods discussed in this chapter. In addition, as discussed in Section 7.5.3, the performance of diversification with cluster ranking is closely related to the performance of the underlying diversification method (that is, without cluster ranking).

[2]The performance of IA-select and RR and their cluster ranking versions is between the median and the best of systems taking part in the diversity task at TREC 2009 Web Track in terms of αNDCG@10 (best: 0.526; median: 0.175) and IA-P@10 (best: 0.244; median: 0.073).

Methods	α-NDCG@5	α-NDCG@10	K	IA-P@5	IA-P@10	K
MRF (baseline)	0.118	0.170	–	0.069	0.088	–
MMR	**0.122**	0.169	–	0.066	0.083	–
cMMR	**0.191**$^{\triangle}$	**0.216**	10	**0.090**	**0.096**	50
FM-LDA	0.027$^{\blacktriangledown}$	0.029$^{\blacktriangledown}$	10	0.011$^{\blacktriangledown}$	0.008$^{\blacktriangledown}$	10
cFM-LDA	0.058$^{\blacktriangledown}$	0.072$^{\blacktriangledown}$	10	0.031$^{\blacktriangledown}$	0.029$^{\blacktriangledown}$	10
IA-select	**0.146**$^{\triangle}$	**0.193**$^{\triangle}$	50	**0.078**	**0.092**	50
cIA-select	**0.199**$^{\blacktriangle}$	**0.221**$^{\triangle}$	10	**0.100**$^{\triangle}$	**0.092**	50
RR	**0.198**$^{\blacktriangle}$	**0.222**$^{\triangle}$	10	**0.079**$^{\blacktriangle}$	0.067$^{\triangledown}$	10
cRR	**0.200**$^{\blacktriangle}$	**0.233**$^{\blacktriangle}$	10	**0.085**$^{\blacktriangle}$	0.083	10

Table 7.5: Performance of the initially retrieved ranked list of documents (MRF) in terms of diversity and the optimal performance of diversification methods and the corresponding cluster ranking versions. Clusters are ranked with query likelihood. Bold face indicates improved performance over the baseline, i.e., MRF. Significance is tested against the MRF baseline.

7.6 Sensitivity analysis

In this section, we offer a first of two rounds of analysis into the effectiveness of diversification with cluster ranking. The analysis in this section provides insights into the sensitivity of our proposed framework to various parameter settings. Specifically, we aim to answer research questions RQ5b and RQ5c.

7.6.1 Impact of the cluster ranker and T

Recall research question RQ5b: What is the impact of the two main components, namely, the cluster ranker and the selection of the number of top ranked clusters, on the overall performance of diversification with cluster ranking? To answer this question, we use a set of oracle experiments based on the oracle cluster ranker; we run the experiments with oracle parameter settings as described in Section 7.4.1.

Figure 7.4 shows the trends of the performance of each diversification method across values of T with the oracle cluster ranker. If we compare Figure 7.4 to Figure 7.3 on page 112, we see that in Figure 7.3 the retrieval performance fluctuates a lot as T increases, that is, with many local maximums, while in Figure 7.4, the performance curves are relatively smooth: they remain the same or decrease once an initial maximum has been reached. This implies that, with a near perfect ranking of clusters, we can find the globally optimal T by simply adding documents belonging to a cluster ranked next, until the performance starts to decrease. On top of that, we clearly see that optimal results are achieved by selecting a small number of top ranked clusters. In addition, we notice that the oracle cluster ranker has a different impact on different diversification methods. For example, in Figure 7.3, cIA-select has a similar performance as cMMR in most cases, while in Figure 7.4, cIA-select consistently

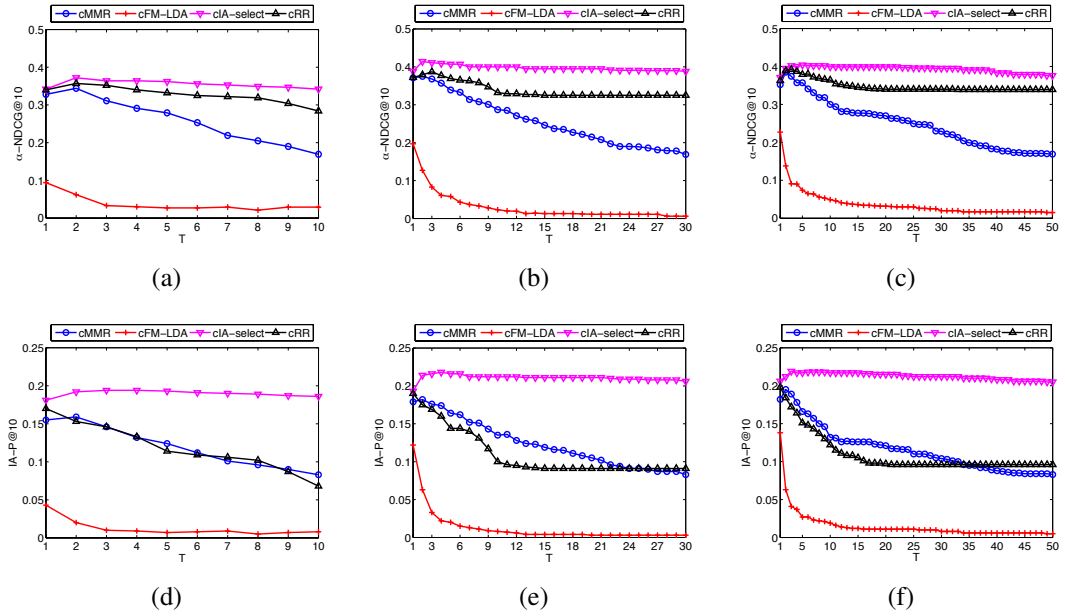


Figure 7.4: Diversification with cluster ranking using oracle information as $cRanker(\cdot)$ over different numbers of selected top ranked clusters (T). The evaluation metrics are α-NDCG@10 (top row) and IA-P@10 (bottom row). The number of clusters K is set to 10 (7.3(a) and 7.3(d)), 30 (7.3(b) and 7.3(e)) and 50 (7.3(c) and 7.3(f)). Note that the plots have different scales at Y-axis for different evaluation metrics.

outperforms other methods.

Now let us take a close look at the results of the oracle experiments, which use oracle information for ranking clusters or determining T, or both. Tables 7.6–7.9 show the oracle performance of diversification in three settings. First, T is selected using an oracle and clusters are ranked with query likelihood. Second, T is automatically determined and the clusters are ranked with the oracle cluster ranker. And, third, both T and the cluster ranker use oracle information. For comparison, we also include results of the following experiments: diversification over the complete ranked list of documents, and diversification with cluster ranking but without oracle information using the predicted T and query likelihood cluster ranker.

Note that for IA-select and RR, since the importance of clusters is taken into account in the original algorithms when ranking with the oracle cluster ranker, their baselines change as well. For the baselines of IA-select and RR with oracle ranker, we use the oracle information to rank the clusters for the two algorithms, but apply diversification on the whole ranked list.

Two observations can be made. First, for each diversification method, both oracle T and the oracle cluster ranker significantly improve the effectiveness of result diversification over their corresponding baselines. Moreover, in the case of automatically determined T, while not all cases are improved over the baselines when using the query

Method	T_o	R_o	α-NDCG@5			α-NDCG@10			IA-P@5			IA-P@10		
MMR	–	–	0.122			0.169			0.066			0.083		
			$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$
cMMR	–	–	**0.191**$^{\triangle}$	**0.157**	**0.178**$^{\triangle}$	**0.216**	**0.171**	**0.214**$^{\triangle}$	**0.070**	**0.077**	**0.090**	0.069	**0.090**	**0.096**
	+	–	**0.281**$^{\blacktriangle}$	**0.284**$^{\blacktriangle}$	**0.264**$^{\blacktriangle}$	**0.313**$^{\blacktriangle}$	**0.307**$^{\blacktriangle}$	**0.311**$^{\blacktriangle}$	**0.140**$^{\blacktriangle}$	**0.147**$^{\blacktriangle}$	**0.144**$^{\blacktriangle}$	**0.138**	**0.138**$^{\blacktriangle}$	**0.146**$^{\blacktriangle}$
	–	+	**0.312**$^{\blacktriangle}$	**0.331**$^{\blacktriangle}$	**0.357**$^{\blacktriangle}$	**0.344**$^{\blacktriangle\blacktriangle}$	**0.344**	**0.385**$^{\blacktriangle}$	**0.146**$^{\blacktriangle}$	**0.212**$^{\blacktriangle}$	**0.204**$^{\blacktriangle}$	**0.142**$^{\blacktriangle}$	**0.178**$^{\blacktriangle}$	**0.195**$^{\blacktriangle}$
	+	+	**0.369**$^{\blacktriangle}$	**0.406**$^{\blacktriangle}$	**0.417**$^{\blacktriangle}$	**0.401**$^{\blacktriangle}$	**0.432**$^{\blacktriangle}$	**0.440**$^{\blacktriangle}$	**0.204**$^{\blacktriangle}$	**0.234**$^{\blacktriangle}$	**0.233**$^{\blacktriangle}$	**0.195**$^{\blacktriangle}$	**0.217**$^{\blacktriangle}$	**0.217**$^{\blacktriangle}$

Table 7.6: Results of MMR, cMMR and the oracle versions of cMMR. For each K and each evaluation metric, the performance of the oracle runs is compared to the corresponding performance of MMR. Columns T_o and R_o list whether the oracle T and R are used. An increased performance compared to the baseline is shown in boldface.

Method	T_o	R_o	α-NDCG@5			α-NDCG@10			IA-P@5			IA-P@10		
			$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$
FM-LDA	–	–	0.027	0.000	0.008	0.029	0.006	0.015	0.011	0.000	0.004	0.008	0.003	0.005
cFM-LDA	–	–	**0.058**	**0.020**$^{\triangle}$	**0.020**	**0.072**$^{\triangle}$	**0.027**$^{\triangle}$	**0.026**	**0.031**$^{\triangle}$	**0.009**$^{\triangle}$	**0.021**$^{\blacktriangle}$	**0.029**$^{\triangle}$	**0.016**$^{\triangle}$	**0.021**$^{\blacktriangle}$
	+	–	**0.081**$^{\blacktriangle}$	**0.036**$^{\blacktriangle}$	**0.069**$^{\blacktriangle}$	**0.092**$^{\blacktriangle}$	**0.044**$^{\blacktriangle}$	**0.077**$^{\blacktriangle}$	**0.041**$^{\blacktriangle}$	**0.018**$^{\blacktriangle}$	**0.032**$^{\blacktriangle}$	**0.035**$^{\blacktriangle}$	**0.025**$^{\blacktriangle}$	**0.034**$^{\blacktriangle}$
	–	+	**0.069**$^{\triangle}$	**0.152**$^{\blacktriangle}$	**0.192**$^{\blacktriangle}$	**0.094**$^{\blacktriangle}$	**0.197**$^{\blacktriangle}$	**0.227**$^{\blacktriangle}$	**0.036**$^{\triangle}$	**0.098**$^{\blacktriangle}$	**0.118**$^{\blacktriangle}$	**0.043**$^{\blacktriangle}$	**0.122**$^{\blacktriangle}$	**0.138**$^{\blacktriangle}$
	+	+	**0.096**$^{\blacktriangle}$	**0.164**$^{\blacktriangle}$	**0.214**$^{\blacktriangle}$	**0.119** $^{\blacktriangle}$	**0.206**$^{\blacktriangle}$	**0.246**$^{\blacktriangle}$	**0.047**$^{\blacktriangle}$	**0.103**$^{\blacktriangle}$	**0.125**$^{\blacktriangle}$	**0.051**$^{\blacktriangle}$	**0.123**$^{\blacktriangle}$	**0.140**$^{\blacktriangle}$

Table 7.7: Results of FM-LDA, cFM-LDA and the oracle versions of cFM-LDA. For each K, the results of the oracle runs are compared to corresponding results of FM-LDA. An increased performance compared to the baseline is shown in boldface.

likelihood-based cluster ranker, the proposed approach outperforms the baselines in all cases when using the oracle cluster ranker, and many of the improvements are statistically significant. That is, the prediction of T is more effective when an oracle cluster ranker is used.

Second, using the oracle cluster ranker results in better performance in terms of the diversification metrics than using an oracle to determine T in all cases except in the case of FM-LDA with $K = 10$ of α-NDCG@5 and IA-P@5; this suggests that the oracle cluster ranker has a larger impact on the diversification results than the oracle T. On top of that, combining the oracle cluster ranker and the oracle T always results in improved performance.

In addition, for methods like IA-select and RR, the oracle information of cluster distribution, as defined in 7.5, helps in both cases, with and without cluster ranking

Method	T_o	R_o	α-NDCG@5			α-NDCG@10			IA-P@5			IA-P@10		
			$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$
IA-select	–	–	0.125	0.116	0.146	0.179	0.165	0.193	0.069	0.063	0.078	0.086	0.073	0.092
cIA-select	–	–	**0.199**$^{\triangle}$	**0.145**	**0.181**$^{\triangle}$	**0.221**	0.158	**0.208**	0.053	**0.079**	**0.100**	0.056	**0.077**	0.092
cIA-select	+	–	**0.287**$^{\blacktriangle}$	**0.252**$^{\blacktriangle}$	**0.262**$^{\blacktriangle}$	**0.317**$^{\blacktriangle}$	**0.285**$^{\blacktriangle}$	**0.291**$^{\blacktriangle}$	**0.153**$^{\blacktriangle}$	**0.130**$^{\blacktriangle}$	**0.137**$^{\blacktriangle}$	**0.150**$^{\blacktriangle}$	**0.123**$^{\blacktriangle}$	**0.127**$^{\blacktriangle}$
IA-select	–	+	0.316	0.362	0.361	0.342	0.388	0.376	0.186	0.212	0.214	0.186	0.206	0.205
cIA-select	–	+	**0.347**$^{\blacktriangle}$	**0.389**$^{\triangle}$	**0.372**	**0.372**$^{\blacktriangle}$	**0.407**	**0.392**	**0.197**$^{\triangle}$	**0.216**	**0.223**	**0.193**$^{\triangle}$	**0.210**	**0.213**
cIA-select	+	+	**0.374**$^{\blacktriangle}$	**0.424**$^{\blacktriangle}$	**0.416**$^{\blacktriangle}$	**0.394**$^{\blacktriangle}$	**0.443**$^{\blacktriangle}$	**0.429**$^{\blacktriangle}$	**0.218**$^{\blacktriangle}$	**0.245**$^{\triangle}$	**0.246**$^{\triangle}$	**0.209**$^{\blacktriangle}$	**0.232**$^{\blacktriangle}$	**0.231**$^{\blacktriangle}$

Table 7.8: Results of IA-select, cIA-select and the oracle versions of cIA-select. For each K, the results of cIA-select and cIA-select with oracle T are compared to that of IA-select, and the cIA-select runs where the oracle cluster ranker is used are compared to the IA-select run with oracle cluster ranker. An increased performance compared to the baseline is shown in boldface.

Method	T_o	R_o	α-NDCG@5			α-NDCG@10			IA-P@5			IA-P@10		
			$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$	$K=10$	$K=30$	$K=50$
RR	–	–	0.198	0.137	0.157	0.222	0.144	0.177	0.079	0.049	0.057	0.067	0.034	0.045
cRR	–	–	**0.199**	**0.151**	**0.160**	**0.233**$^{\triangle}$	**0.168**	0.172	**0.085**	**0.065**	**0.067**	**0.083**	**0.060**$^{\triangle}$	**0.056**
cRR	+	–	**0.225**	**0.197**$^{\blacktriangle}$	**0.202**$^{\blacktriangle}$	**0.274**$^{\triangle}$	**0.230**$^{\blacktriangle}$	**0.243**$^{\blacktriangle}$	**0.107**$^{\triangle}$	**0.095**$^{\blacktriangle}$	**0.096**$^{\blacktriangle}$	**0.125**$^{\blacktriangle}$	**0.085**$^{\blacktriangle}$	**0.096**$^{\blacktriangle}$
RR	–	+	0.296	0.334	0.346	0.284	0.325	0.339	0.121	0.146	0.153	0.068	0.091	0.096
cRR	–	+	**0.339**$^{\blacktriangle}$	**0.362**$^{\blacktriangle}$	**0.361**	**0.357**$^{\blacktriangle}$	**0.387**$^{\blacktriangle}$	**0.384**	**0.179**$^{\blacktriangle}$	**0.205**$^{\blacktriangle}$	**0.202**$^{\blacktriangle}$	**0.170**$^{\blacktriangle}$	**0.190**$^{\blacktriangle}$	**0.198**$^{\blacktriangle}$
cRR	+	+	**0.382**$^{\blacktriangle}$	**0.413**$^{\blacktriangle}$	**0.422**$^{\blacktriangle}$	**0.409** $^{\blacktriangle}$	**0.442**$^{\blacktriangle}$	**0.442**$^{\blacktriangle}$	**0.223**$^{\blacktriangle}$	**0.239**$^{\blacktriangle}$	**0.239**$^{\blacktriangle}$	**0.208**$^{\blacktriangle}$	**0.225**$^{\blacktriangle}$	**0.226**$^{\blacktriangle}$

Table 7.9: Results of RR, cRR and the oracle versions of cRR. For each K, the results of cRR and cRR with oracle T are compared to that of RR, and the cRR runs where the oracle cluster ranker is used are compared to the RR run with oracle cluster ranker. An increased performance compared to the baseline is shown in boldface.

and selection, as these methods take into account the importance of clusters and oracle information provides a good approximation of the importance of clusters. In the case of MMR and FM-LDA, where importance of clusters is not considered, oracle information of cluster distribution only helps when cluster ranking is applied.

In summary, as an answer to research question RQ5b, we find that both the cluster ranker and the cut-off value T are important for the effectiveness of our proposed diversification with cluster ranking framework. Oracle information for either the cluster ranker or the cut-off value T, or both, improves the performance of the proposed framework. This indicates that the performance of each component has a large impact on the overall performance of our framework. The cluster ranker has a larger impact than the cut-off value T on the effectiveness of the proposed framework.

7.6.2 Length effect

Now we turn to research question RQ5c: Given that we use top ranked documents retrieved in response to a query for clustering as well as for diversification, how sensitive is the performance of the proposed framework to the length of the list of documents being selected?

We conduct the analysis experiments as described in Section 7.4.1, where we vary the number of documents for clustering and for applying Algorithm 1 in three settings. In order to summarize the massive amount of experimental results generated by the three settings along with variations of other parameters, such as the number of clusters K, the diversification method used ($Div(\cdot)$) and the number of top ranked clusters selected (T), we use the following three types of scores: Min, Max and Avg. Specifically, for a given experimental setting, a given K and a given $Div(\cdot)$, we apply Algorithm 1 with all possible values of $T \in \{1,...,K\}$ with the oracle cluster ranker. For simplicity, we only use α-NDCG@10 as the evaluation metric. Then for each T we evaluate the results as the average α-NDCG@10 scores over all 50 queries. If we write the evaluation result as $E(T)$, i.e., as a function of T, we have

$$\text{Min} = \arg\min_T E(T), \quad \text{Max} = \arg\max_T E(T), \quad \text{and} \quad \text{Avg} = \sum_T E(T)/K.$$

In other words, we compare the results from different settings in their worst performance, best performance and average performance under different values of T, in terms of α-NDCG@10 which is averaged over 50 queries. For each setting, as described in Section 7.4.1, we compare the results of different settings of C and D, i.e., number of documents used for training the LDA model and the number of documents used for applying Algorithm 1, respectively, to the result of our baseline setting, i.e., $C = 500$ and $D = 1000$. We use two-sided paired t-test for significance test, where the significance level is set to 0.05. Tables 7.10–7.12 show the results.

In Table 7.10 we see that, in general, the differences between different settings of C are not significant. There are two exceptions: cRR with $K = 50$, $C = 100$, which significantly outperforms $C = 500$ in all three types of scores; and cFM-LDA with $K = 30$, $C = 100$, where the performance difference is significant in terms of Avg scores. However, these occasional significant differences in performance may due be to various reasons; no clear pattern emerges in the overall performance when using different numbers of document for training the LDA models under our diversification framework.

From Table 7.11 we make two observations. First, we see that in general, smaller values of D (i.e., $D = 100, 300$) are preferred to $D = 1000$, as in all cases, none of the $D = 1000$ outperform their $D = 100, 300$ counterparts in terms of absolute values of evaluation score. Second, for each diversification method, we see certain patterns in their performance with different settings of D. For cMMR, cFM-LDA and cRR, in general, $D = 100$ is preferred, as it achieves best performance in 24 out of 27 cases. Particularly, in terms of Max scores, for all 3 diversification methods, $D = 100$ results in best performance. Also, we see that for cFM-LDA, all the differences between the $D = 100, 300$ and $D = 1000$ are statistically significant. On the other hand, cIA-select is an interesting exception among other diversification methods: it does not show significant difference between different settings of D in any of cases. However, cIA-select seems to slightly prefer $D = 300$, as it results in best scores for all cases.

In Table 7.12 we see a similar pattern as in Table 7.10 for cMMR and cFM-LDA. That is, small numbers of documents ($C = 100, D = 100$ and $C = 300, D = 300$) are preferred over a large number of documents ($C = 500, D = 1000$). In addition, the observation that significant differences between different settings of C and D occur under similar conditions as in Table 7.11 suggests that the results of Setting 3 are an effect of D, the number of documents to which Algorithm 1 is applied. Besides, cIA-select still shows no significant difference between different settings of C and D, with a slight preference towards $C = 300, D = 300$.

In summary, we have the following conclusions for answering research question RQ5c. We find that the number of documents used for clustering does not have a significant and systematic impact on the overall performance of our proposed diversification framework. On the other hand, the number of documents used for applying Algorithm 1 displays a systematic impact on the overall performance of the proposed diversification framework. For all diversification methods, a smaller number of documents, e.g., 100, 300, is preferred over a large number, which is set to 1000 in our

		$K = 10$			$K = 30$			$K = 50$		
Method	Score	C100	C300	C500	C100	C300	C500	C100	C300	C500
cMMR	Min	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**	**0.169**
	Max	**0.351**	0.340	0.344	0.379	**0.399**	0.375	**0.405**	0.403	0.385
	Avg	0.258	**0.262**	0.259	**0.267**	0.256	0.256	**0.258**	0.257	0.248
cFM-LDA	Min	0.025	**0.039**	0.021	**0.022**	0.011	0.006	0.011	0.008	**0.015**
	Max	**0.115**	0.097	0.094	0.226	**0.237**	0.197	0.269	**0.271**	0.227
	Avg	0.050	**0.053**	0.038	**0.056**$^{\triangle}$	0.036	0.031	**0.041**	0.031	0.038
cIA-select	Min	0.334	0.334	**0.342**	0.389	**0.405**	0.388	0.386	**0.407**	0.371
	Max	0.369	0.371	**0.372**	0.407	**0.425**	0.414	0.409	**0.431**	0.404
	Avg	0.347	0.350	**0.355**	0.396	**0.416**	0.397	0.393	**0.414**	0.393
cRR	Min	0.283	0.281	**0.284**	0.351	**0.357**	0.325	**0.372**$^{\triangle}$	0.356	0.339
	Max	0.358	**0.367**	0.357	0.395	**0.409**	0.387	**0.423**$^{\triangle}$	0.416	0.392
	Avg	0.324	**0.328**	0.328	0.361	**0.370**	0.339	**0.379**$^{\triangle}$	0.367	0.348

Table 7.10: Results of Setting 1: $C = 100, 300, 500$; $D = 1000$. In each block, scores from columns $C100$ and $C300$ are compared to their corresponding scores in column $C500$; statistically significant difference between the scores from $C100$ ($C300$) and that from $C500$ is annotated by $^{\triangle}$. The highest scores among different settings of C for a given K, a given diversification method and a given type of score (Min, Max or Avg) are shown in boldface.

experiments. In addition, we find that for cIA-select, both parameters do not show significant impact on the final diversification results.

7.7 Impact of clustering structure

Now let us turn to research question RQ5d: What conditions should clusters fulfill in order for diversification with cluster ranking to be effective?

Since our prime motivation for applying query-specific clustering and cluster ranking to result diversification is its effect on promoting relevance, we first examine the type of properties that makes query-specific clustering effective in promoting precision. On the other hand, from a diversification perspective, we expect that documents contained in those top ranked clusters, while relevant to the general topic of a given query, cover multiple facets or sub-topics of the general topic. Intuitively, if the documents contained in the top ranked clusters exclusively focus on a single narrow topic, diversification will not be effective due to the lack of diverse content.

In summary, we expect that the clusters generated by a query-specific clustering algorithm should satisfy the following conditions to make diversification with cluster ranking effective:

Condition 1 Among all clusters, there exist a small number of clusters, which we call *high quality clusters*, that contain most of the *relevant* documents;

Condition 2 The union of *high quality clusters* should contain documents associated

		$K=10$			$K=30$			$K=50$		
Method	Score	D100	D300	D1000	D100	D300	D1000	D100	D300	D1000
cMMR	Min	**0.171**	0.169	0.169	**0.171**	0.169	0.169	**0.171**	0.169	0.169
	Max	**0.378**$^{\triangle}$	0.361	0.344	**0.422**$^{\triangle}$	0.416$^{\triangle}$	0.375	**0.405**	0.385	0.385
	Avg	0.260	**0.262**	0.259	0.254	**0.259**	0.256	**0.252**	0.250	0.248
cFM-LDA	Min	**0.132**$^{\triangle}$	0.089$^{\triangle}$	0.021	**0.095**$^{\triangle}$	0.027$^{\triangle}$	0.006	**0.099**$^{\triangle}$	0.042$^{\triangle}$	0.015
	Max	**0.341**$^{\triangle}$	0.237$^{\triangle}$	0.094	**0.395**$^{\triangle}$	0.340$^{\triangle}$	0.197	**0.357**$^{\triangle}$	0.320$^{\triangle}$	0.227
	Avg	**0.198**$^{\triangle}$	0.132$^{\triangle}$	0.038	**0.157**$^{\triangle}$	0.074$^{\triangle}$	0.031	**0.144**$^{\triangle}$	0.076$^{\triangle}$	0.038
cIA-select	Min	0.327	**0.352**	0.342	0.359	**0.392**	0.388	0.366	**0.385**	0.371
	Max	0.369	**0.374**	0.372	0.422	**0.432**	0.414	0.402	**0.404**	**0.404**
	Avg	0.347	**0.362**	0.355	0.376	**0.407**	0.397	0.381	**0.396**	0.393
cRR	Min	**0.315**$^{\triangle}$	0.287	0.284	0.341	**0.344**	0.325	**0.341**	0.331	0.339
	Max	**0.376**	0.363	0.357	**0.425**$^{\triangle}$	0.418$^{\triangle}$	0.387	**0.421**	0.403	0.392
	Avg	**0.349**	0.333	0.328	**0.368**$^{\triangle}$	0.362$^{\triangle}$	0.339	**0.369**	0.347	0.348

Table 7.11: Results of Setting 2: $C = 500$, $D = 100, 300, 1000$. In each block, scores from columns $D100$ and $D300$ are compared to their corresponding scores in column $D1000$; statistically significant difference between scores from $D100$ ($D300$) and that from $D1000$ is annotated by $^{\triangle}$. The highest scores among different settings of C for a given K, a given diversification method and a given type of score (Min, Max or Avg) are shown in boldface.

		$K=10$			$K=30$			$K=50$		
Method	Score	top100	top300	top500	top100	top300	top500	top100	top300	top500
cMMR	Min	**0.171**	0.169	0.169	**0.171**	0.169	0.169	**0.171**	0.169	0.169
	Max	**0.381**$^{\triangle}$	0.363	0.344	**0.434**$^{\triangle}$	0.430$^{\triangle}$	0.375	0.411	**0.424**	0.385
	Avg	**0.270**	0.266	0.259	0.259	**0.261**	0.256	0.241	**0.255**	0.248
cFM-LDA	Min	**0.101**$^{\triangle}$	0.069$^{\triangle}$	0.021	**0.055**$^{\triangle}$	0.040$^{\triangle}$	0.006	**0.050**$^{\triangle}$	0.037	0.015
	Max	**0.327**$^{\triangle}$	0.225$^{\triangle}$	0.094	**0.398**$^{\triangle}$	0.363$^{\triangle}$	0.197	**0.355**$^{\triangle}$	0.354$^{\triangle}$	0.227
	Avg	**0.171**$^{\triangle}$	0.107$^{\triangle}$	0.038	**0.123**$^{\triangle}$	0.082$^{\triangle}$	0.031	**0.094**$^{\triangle}$	0.072$^{\triangle}$	0.038
cIA-select	Min	0.342	**0.348**	0.342	0.382	**0.407**	0.388	0.381	**0.395**	0.371
	Max	0.376	**0.386**	0.372	0.420	**0.440**	0.414	0.415	**0.427**	0.404
	Avg	0.360	**0.362**	0.355	0.400	**0.428**	0.397	0.390	**0.409**	0.393
cRR	Min	**0.303**	0.275	0.284	**0.370**	0.363	0.325	0.328	**0.346**	0.339
	Max	**0.386**	0.368	0.357	**0.438**$^{\triangle}$	0.436$^{\triangle}$	0.387	0.408	**0.439**	0.392
	Avg	**0.349**	0.328	0.328	**0.394**$^{\triangle}$	0.380$^{\triangle}$	0.339	0.356	**0.367**	0.348

Table 7.12: Results of Setting 3: $top100$ denotes $C = 100, D = 100$, $top300$ denotes $C = 300, D = 300$ and $top500$ denotes $C = 500, D = 1000$. In each block, scores from columns $top100$ and $top300$ are compared to their corresponding scores in column $top500$; statistically significant difference between scores from $top100$ ($top300$) and that from $top500$ is annotated by $^{\triangle}$. The highest scores among different settings of C for a given K, a given diversification method and a given type of score (Min, Max or Avg) are shown in boldface.

with multiple facets of a query, or in other words, documents whose content are sufficiently different.

In the following sections, we examine the impact of the above two conditions on the effectiveness of our diversification with cluster ranking framework.

7.7.1 Preliminaries

Measuring the two conditions

In order to examine how the two conditions mentioned above are reflected by different types of clustering structures, we need measures that are able to capture the characteristics of a given clustering structure with respect to these two conditions.

Note that in Section 6.3.4 on page 87, we have already studied the properties of clustering structure that promote precision, which is exactly the requirement stated by Condition 1. Here we translate Condition 1 into the Precision score, which on the one hand, measures the amount of relevant documents contained in a given set of documents, and on the other hand, limits the size of the set of documents. That is, we do not want to have a set of clusters containing most of the relevant documents merely due to the fact that most documents are assigned to them.

For Condition 2, we propose to use an adapted version of the Coherence Score, which reverses the score so as to reflect "diversity" instead of "coherence." It is defined as:

$$rCoh(D) = 1 - Co(D). \tag{7.6}$$

As pointed out in Chapter 3, the coherence score gives a higher value to a structured data set than to a random set, and among structured data sets it gives higher values to sets with fewer clusters. In our case, the reversed Coherence score gives a high score to a set of documents if it has a rich sub-cluster structure; a low score if documents within the set are highly similar.

7.7.2 Clustering structure

Now let us look at the clustering structure generated by the LDA models and hierarchical clustering with complete linkage, in terms of Precision and reversed Coherence scores.

Note that in Algorithm 1, given a ranked list of clusters, the diversification procedure is applied to the union of the documents contained in the top T clusters. Accordingly, the Precision and reversed Coherence scores are also calculated on the union of documents belonging to the top T clusters, which we refer to as accumulated Precision and accumulated reversed Coherence Scores, as the measures are taken on all the documents in the selected clusters.

To illustrate the cluster structure with respect to Condition 1, we first rank the clusters using the oracle cluster ranker as described in 7.3.2, which is equivalent to

ranking with accumulated Precision scores. Then we plot the distribution of the accumulated Precision scores and reversed Coherence Scores for documents in the top T clusters, where $T = 1, ..., K$. Figure 7.5 shows the distribution of accumulated Precision scores and Figure 7.6 shows the distribution of accumulated reversed Coherence Scores among documents from the top T clusters. We see an interesting difference between the two clustering algorithms, namely, LDA and hierarchical clustering with complete linkage.

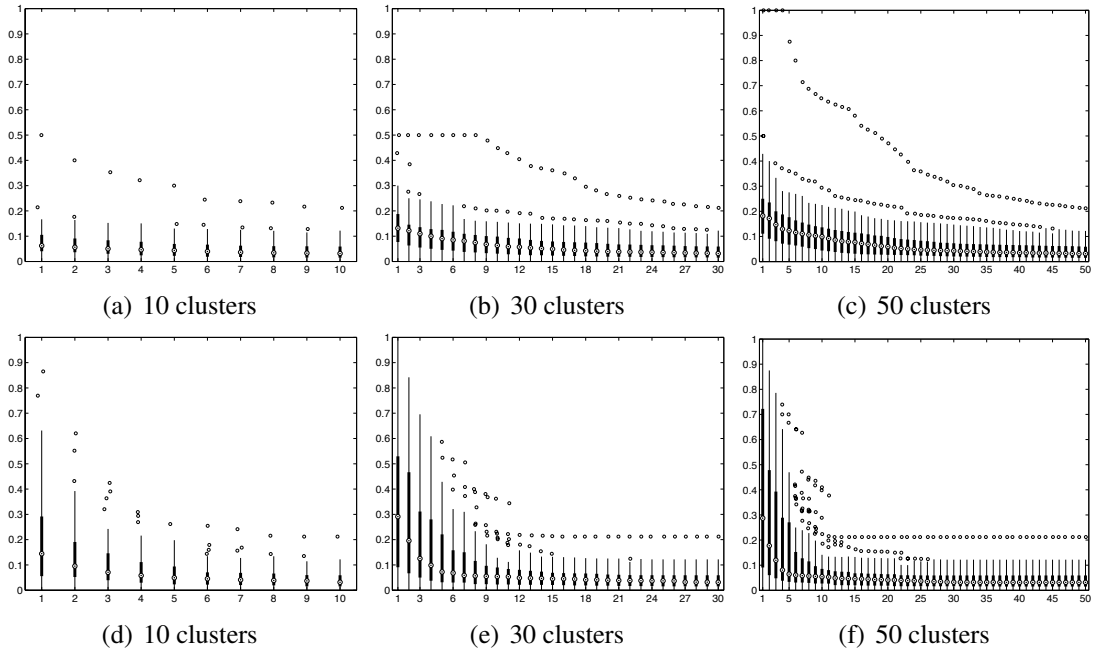
In Figure 7.5 we see that the early Precision scores of clusters generated by LDA are higher than those generated by the hierarchical clustering on average, but also have a larger variance. Note that the accumulated Precision score for the two clustering algorithms should converge to the same value at some point, as the same initial ranked list is used for both clustering procedures. For LDA, as the number of clusters being included increases, the accumulated Precision scores decrease quickly, while for hierarchical clustering, the change is not very obvious, especially in the case of 10 clusters. The above observations suggest that clusters generated by LDA are more likely to satisfy Condition 1 than clusters generated by hierarchical clustering.

In Figure 7.6 we see that top ranked clusters generated by hierarchical clustering with complete linkage have higher accumulated reversed Coherence Scores than those generated by LDA. In other words, the clusters generated by LDA are more likely to focus on a single topic or just a few sub-topics, while clusters generated by hierarchical clustering are more likely to contain documents associated with multiple sub-topics or with diverse content. These observations suggest that clusters generated by hierarchical clustering are more likely to satisfy Condition 2 than those generated by LDA, that is, containing more diverse material.

7.7.3 Impact on the performance of the proposed diversification framework

Now that we have seen that the clusters generated by LDA and hierarchical clustering have a difference in their clustering structure, let us examine whether this difference has an impact on the overall performance of our proposed diversification framework.

Figure 7.7 shows the results of diversification with cluster ranking with hierarchical clustering and LDA, in terms of α-NDCG@10 and IA-P@10. All clusters are ranked with the oracle cluster ranker, so that we see how the clustering structure influences the performance under a perfect ranking. To incorporate hierarchical clustering into our proposed diversification framework, for cRR and cMMR, we simply apply Algorithm 1 with the clusters generated by hierarchical clustering. For cFM-LDA and cIA-select, we use hierarchical clustering to generate the clusters, and select the top ranked clusters for diversification. While applying Algorithm 1, we still use LDA for modeling the sub-topics of a query. That is, hierarchical clustering is only used for selecting documents to be diversified. In addition, in Table 7.13 we show the Pearson correlation between the end performance of our proposed framework and the Precision



(a) 10 clusters (b) 30 clusters (c) 50 clusters

(d) 10 clusters (e) 30 clusters (f) 50 clusters

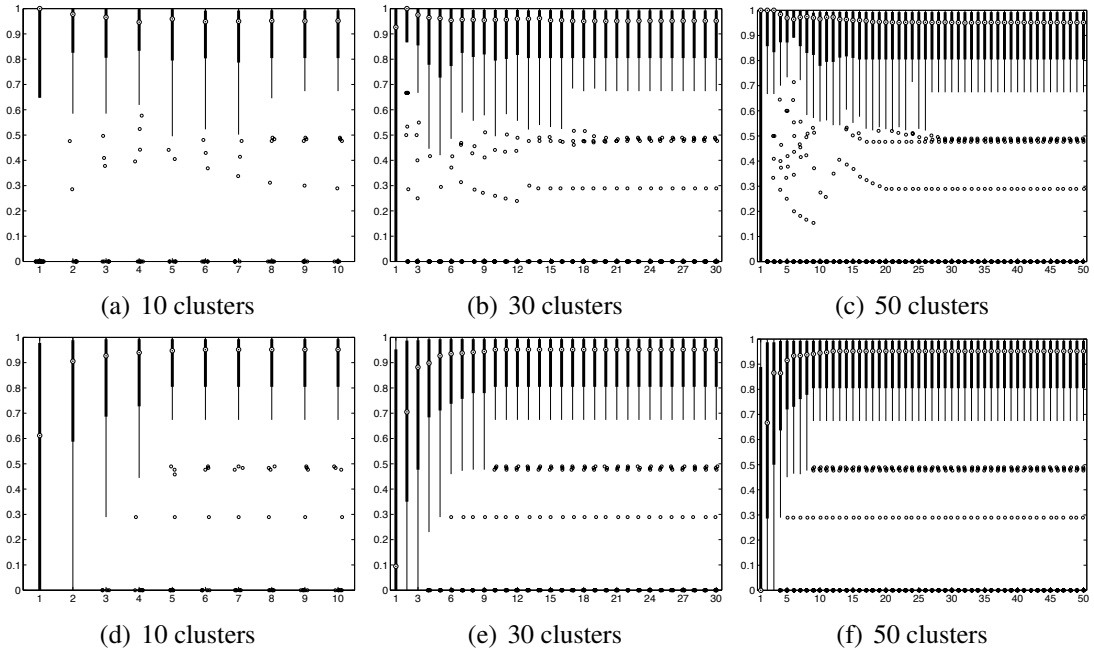
Figure 7.5: Distribution of accumulated Precision scores among clusters. Figures 7.5(a)–7.5(c) show the accumulated precision scores for clusters generated by hierarchical clustering, over 50 queries. Figures 7.5(d)–7.5(f) show the same scores for clusters generated by LDA. In each box, the "$\odot$" at the central position is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "$\circ$".

scores and reversed Coherence scores, which are calculated as in the previous section (Section 7.7.2). All correlations are significant except in the case for cFM-LDA, where the correlation between the reversed Coherence score and diversification result is not significant.

We notice that different diversification methods show different behaviors given different clustering algorithms. Let us refer to a diversification with cluster ranking procedure based on LDA as "the LDA version," and a procedure based on hierarchical clustering with complete linkage as "the HC version."

For cMMR and cFM-LDA, we see that from Figure 7.7 that initially, the LDA versions outperform their corresponding HC versions in all three settings of K, number of clusters, set to 10, 30 and 50. As T increases, the HC versions can outperform the LDA versions, and vice versa; when $T = K$, since both versions are applied on the same initial ranked list, the performance ends up as the same.

In Table 7.13 we see that for cMMR and cFM-LDA, the correlation scores between the diversification results (measured by α-NDCG@10 and $IA-P$@10) and the Precision score is stronger than that between the diversification results and the reversed



(a) 10 clusters (b) 30 clusters (c) 50 clusters

(d) 10 clusters (e) 30 clusters (f) 50 clusters

Figure 7.6: Distribution of accumulated reversed Coherence scores among clusters. Figures 7.6(a)–7.6(c) show the accumulated reversed Coherence scores for clusters generated by hierarchical clustering, over 50 queries. Figures 7.6(d)–7.6(f) show the same scores for clusters generated by LDA. In each box, the "$\odot$" at the central position is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually as "$\circ$".

Coherence scores. This may be the reason why the initial performance of the LDA versions is better than that of the HC versions. As with small T, the early precision of top ranked clusters has a larger impact on the performance of the proposed diversification framework. Recall that in Section 7.5 we noticed that the cFM-LDA selects relatively small T and we hypothesized that cFM-LDA is very sensitive to the non-relevant documents included when more clusters are included for diversification. The high correlation between the performance of cFM-LDA and the Precision scores, as we see from Table 7.13, further suggest that the gain of cFM-LDA by applying diversification with cluster ranking comes from the increased precision at the top ranked clusters.

For cRR and cIA-select, we see that in Table 7.13, α-NDCG@10 is found to have a stronger correlation with the reversed Coherence scores than with the Precision scores, while IA-P@10 has a stronger correlation with the Precision scores than with the reversed Coherence scores. In Figure 7.7, correspondingly, we see that for IA-P@10, the LDA versions greatly outperform the HC versions at small Ts, which may be caused by the high early precision of the LDA versions. For α-NDCG@10, where the correla-

tion between the diversification performance and the precision is not as strong, we see that for cRR, the LDA versions only slightly outperform the HC versions for small Ts and for cIA-select, the HC versions outperform the LDA versions. We also notice that for these two diversification methods, for larger Ts in the case of $K = 30$ and 50, the HC versions outperform the LDA versions in terms of both evaluation metrics, which suggests that the HC version may have achieved a better balance between Condition 1 and Condition 2 than the LDA version at larger Ts for these two methods.

Finally, it seems that out evaluation measures, α-NDCG and IA-P, have different preferences concerning relevance and diversity. In particular, IA-P has a bias towards precision as it consistently has a higher correlation with precision than with reversed coherence.

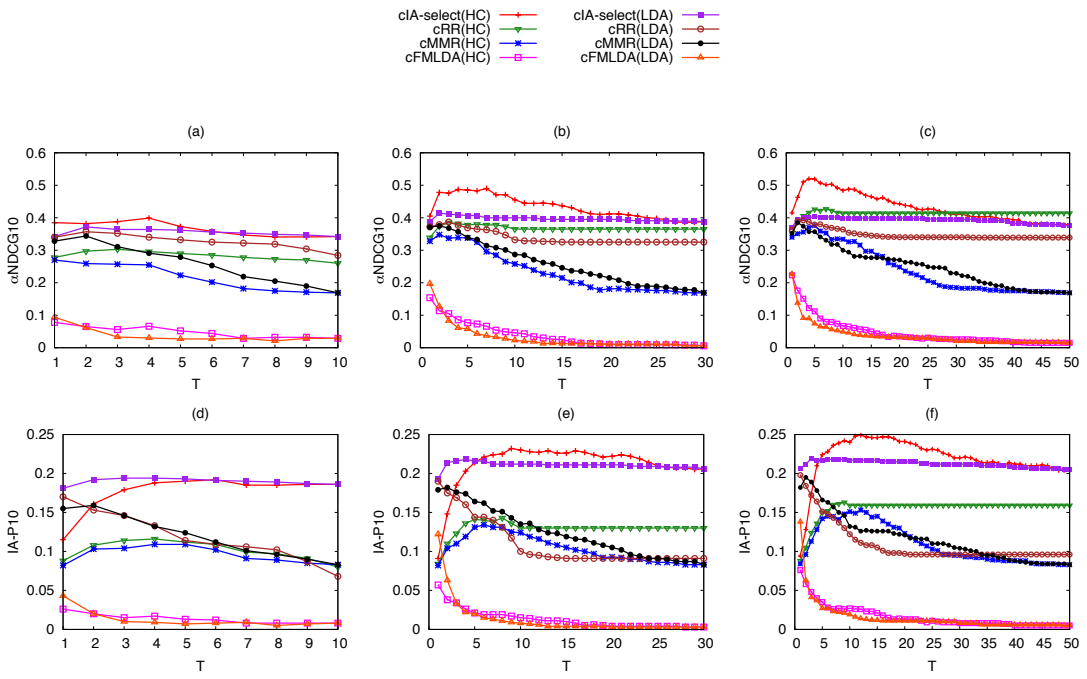


Figure 7.7: Comparison of diversification results using the clusters generated by hierarchical clustering to clusters generated by LDA. In both cases, the clusters are ranked by oracle cluster ranker.

7.7.4 Conclusions

In summary, in this section, we posit that the clusters generated by a clustering algorithm should fulfill two conditions with respect to precision and diversity for our proposed diversification framework to be effective. Empirical results show that for most diversification methods, both conditions are significantly correlated with the overall performance of the framework. The impact of the two conditions on the overall per-

Measure	Comp. Type	cRR	cIA-select	cFM-LDA	cMMR
αNDCG@10	Precision	0.2968	0.2384	0.6117	0.3862
	rCoh	0.4062	0.2714	0.0549	0.3212
IA-P@10	Precision	0.4180	0.4135	0.6863	0.3612
	rCoh	0.2391	0.2528	$0.0133^{\ddagger}$	0.1650

Table 7.13: Pearson correlation coefficients. All correlation scores are statistically significant (p-value < 0.01) except the one with the ‡ sign. Precision refers to the accumulated Precision scores for top T clusters, where $T = 1, ..., K$ and rCoh refers to the accumulated reversed Coherence scores.

formance, however, is dependent on the type of diversification method used, which suggests that when choosing a specific clustering algorithm, one should take into account the properties of the diversification method to be used.

7.8 Conclusions and further discussions

We investigated whether and how query-specific clustering can be used for improving the effectiveness of result diversification. More specifically, our aim was to take advantage of cluster-based retrieval methods for promoting relevance and restricting result diversification to a select set of high quality clusters that contain large numbers of relevant documents so as to improve the effectiveness of diversification in terms of both relevance and diversity.

Our main findings can be summarized as follows. First, we proposed a diversification framework based on query-specific clustering with cluster ranking and selection, in which the diversification procedure is restricted to documents associated with clusters that potentially contain large numbers of relevant documents. The framework was shown to improve the performance, as measured by α-NDCG and IA-P, of several types of diversification methods using a query likelihood based cluster ranker and a cluster cut-off value T which is automatically determined via cross-validation.

On top of that, we analyzed the effectiveness of the proposed diversification framework with respect to four aspects: the cluster rankers, the cluster cut-off value T, the length effect of the initially retrieved ranked list, as well as the clustering structure generated by the clustering algorithms. We showed that both the performance of the cluster ranker and the choice of the cluster cut-off value T are crucial to the overall performance of our diversification framework. Also, the overall performance of the proposed framework is influenced by the length of the initially ranked list of documents. We posited two conditions that the clusters generated by a clustering algorithm should fulfill in order for the diversification with cluster ranking to be effective. Our empirical results have shown that these conditions have a strong correlation with the overall performance, but the strength of the impact of each condition depends on the specific diversification method that is used. In addition, the question of "which cluster-

ing algorithm can effectively generate the desired clustering structure" remains, which we leave for the future work.

Our findings are interesting for the development of new diversification methods as well as for cluster-based retrieval models for faceted queries. At the same time, various options for further analyses within our proposed diversification framework remain. We have only experimented with a simple strategy for ranking clusters and the oracle experiments show that there is sufficient room for improvement with more sophisticated ranking approaches. Similarly, we have shown that there exists an optimal value of T, with which the effectiveness of diversification can be maximized. Clearly, more sophisticated learning methods should be explored for this purpose.

Conclusion to Part II

In this part of the thesis, we addressed the second research theme: *diversity and the cluster hypothesis*. Specifically, in Chapter 6 we re-visited the cluster hypothesis with respect to ambiguous and multi-faceted queries. On top of that, in Chapter 7 we proposed a result diversification approach based on query-specific clustering and cluster ranking.

Our main findings can be summarized as follows. First, we found that with respect to ambiguous and multi-faceted queries, the cluster hypothesis is valid. We empirically validated the cluster hypothesis in two ways: (i) in terms of inter-document similarities, for a given query, relevant documents tends to be more similar to each other than to non-relevant documents; and (ii) in terms of topical coherence, for a given query, document sets consisting of only relevant documents are topically more coherent than document sets consisting of both relevant and non-relevant documents, as measured by the coherence score. It is worth mentioning, however, that the above two statistics, i.e., the distribution of inter-document similarities and the coherence scores, do exhibit difference for ambiguous and multi-faceted queries compared to that of specific or single-faceted queries.

Second, empirical results on the TREC 2009 Web test collection show that the proposed framework improves the performance of several existing diversification methods, including MMR, IA-select and FM-LDA. The framework also gives rise to a simple yet effective cluster-based approach to result diversification that selects documents from different clusters to be included in a ranked list in a round robin fashion. We described a set of experiments aimed at thoroughly analyzing the behavior of the main components of the proposed diversification framework, including ranking and selecting clusters for diversification, and the query-specific clustering structure desired by our proposed diversification framework.

Our findings in this part of the thesis provide renewed insights into the relation between retrieval effectiveness and the cluster hypothesis, as well as implications for future work in result diversification.

Part III

Relating Topics in Different Representations

Chapter 8

Automatic Link Generation with Wikipedia

In Part I and Part II of the thesis we have been using an implicit and internal representation of topics. That is, topics are represented using the statistics of the terms within the documents being analyzed. In part III, we turn to an explicit and external representation, and zoom in on the word/phrase level to look at topic representations. Using definition or descriptions from an external knowledge base to represent the topical information identified in a word/phrase in context. Such a representation is useful for providing background knowledge that helps users to understand difficult concepts as well as to capture the meaning of ambiguous words or phrases while reading a piece of text. In this part of the thesis, we focus on how this type of representation can be established. We study this problem in the context of Automatic Link Generation (ALG) with Wikipedia, which can be described as follows: for a given piece of text, which is referred to as a *source text*, identify a set of *anchor texts*, i.e., words and phrases that need background information from a knowledge base, and for each *anchor text*, find a *target page* in Wikipedia that provides the background information for it. In short, there are two problems that need to be solved in the ALG task, namely, for each word/phrase in a source text, (i) whether a link should be generated? and (ii) if so, where to link to?

As discussed in Chapter 2, both as a target knowledge base as well as a training collection, Wikipedia provides useful statistics and that has been exploited by many studies. Particularly, the data-driven approach proposed by Milne and Witten [178] that uses existing Wikipedia links as training examples has shown state-of-the-art performance. Here, we study the problem of "learning to link with Wikipedia" from two perspectives.

In Chapter 8, we analyze the impact of a set of factors on the performance of automatic link generation while learning linking patterns from Wikipedia itself. More specifically, what are the impact of training collections and learning methods? In addition, is the model learnt from existing Wikipedia links effective when evaluated with manual assessments? In summary, we formulate the following main research question for this chapter:

RQ6 While exploring Wikipedia's link structure for relating the two topical representations, what is the impact of the evaluation type, training collection and learning methods?

In Chapter 9, we move from using Wikipedia for evaluation towards a more realistic problem, where we aim to annotate radiology reports by generating links from medical phrases in the reports to Wikipedia. The main research question we address in Chapter 9 is as follows.

RQ7 Can state-of-the-art ALG systems that are, in principle, domain independent, be effectively applied to linking texts from a specific domain to Wikipedia? If not, can we improve the effectiveness of automatic link generation by considering domain specific properties of the data?

8.1 Introduction

In this chapter, we focus on exploring machine learning methods and learning material for link detection. We conduct this study within the context of the link-the-wiki task specified at INEX. The main purpose of our study here is as follows.

First, we want to test how our learning methods work on the link-the-wiki task. Particularly, Huang et al. [113] have shown that existing Wikipedia links are far from perfect when evaluated against human assessments: there exist many trivial links such as dates in the Wikipedia links, which are actively rejected by human assessors. Here we are interested in how the results learnt from the existing Wikipedia links will be judged by human assessors. On top of that, as discussed in Section 2.3.3 on page 23, within the context of the link-the-wiki task, automatic link generation is defined as a ranking problem for recommendation purposes, we are interested in how a learning to rank approach works as it directly optimizes the rankings instead of assigning binary decisions to candidate links as a classification method would do. Specifically, we formulate two different learning problems, i.e., a binary classification problem versus a ranking problem. Both problems are solved using Support Vector Machines (SVMs) [47]. See Section 8.2.2 for details about the learning problems and how SVMs are used to solve them.

Second, we train our models with different versions of Wikipedia. The two versions used, namely Wikipedia 2008 [56] and Wikipedia 2009 [219], differ in the amount of articles they contain as well as in the amount of links, as pages are added and deleted as time passes by. We experiment with both collections so as to see the impact of the training material used.

In addition, we explore a set of features for constructing the classifiers/rankers. In order to examine the effectiveness of the features, we also heuristically combine the two intuitively most useful features without sophisticated learning methods.

In summary, we seek answers to the following specific research questions (with respect to our main research question RQ6 introduced above):

RQ6a When the ALG task is viewed as a ranking problem, is a learning to rank approach more effective than a binary classification approach?

RQ6b Do different versions of the Wikipedia collection (with, potentially, differences in collection size, numbers of links, etc.) result in performance differences when used as training material?

RQ6c Are the features used for learning the models effective? Are there single features whose contribution to the linking results is dominant?

The rest of the chapter is organized as follows. In Section 8.2 we first specify the notation we use throughout this part of the thesis, followed by a brief introduction to using SVM for binary classification and for ranking. In Section 8.3 we introduce the learning approaches applied to our task. We specify the experimental setup in Section 8.4. In Section 8.5 we discuss the experimental results. We conclude in Section 8.6.

8.2 Preliminaries

8.2.1 Notation

Let $T = \{t_i\}_{i=1}^{|T|}$ be a set of *source texts*, and $W = \{d_j\}_{j=1}^{|W|}$ be the Wikipedia collection, where $|\cdot|$ denotes the number of elements in a set.

The goal of a link generation system is to (i) identify a set of anchor texts $A^t = \{a_k\}_{k=1}^{|A^t|}$ from t, e.g., a radiology report, and for each a, (ii) find a *target* page $d^* \in W$ such that a and d^* form a link $l(a, d^*)$. We refer to the first task as *anchor text detection*, and the second task as *target finding*.

The anchor text a is selected from a set of all possible ngrams $NG^t = \{ng_n\}_{n=1}^{|NG^t|}$ found in t. The target page d^* is selected from a set of *candidate target pages* for anchor a, which is written as $C^a = \{c_m\}_{m=1}^{|C^a|}$. The candidate target set can be the whole Wikipedia collection, but in practice, often a subset of the Wikipedia collection is considered for efficiency reason.

8.2.2 SVM: binary classification versus ranking

In this section, we briefly review using Support Vector Machines (SVMs) for solving a binary classification problem and for solving a ranking problem. In the next section, we will use these two algorithms to solve our learning problem in the context of ALG.

Binary classification

Let a set of training instances $D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \Re^M, y_i \in \{-1, 1\}\}_{i=1}^{N}$ be given, where $\mathbf{x}_i$ is an M-dimensional vector, e.g., an instance with M features and y_i is the corresponding label of $\mathbf{x}_i$, which is either -1 or $+1$. The goal of the classification algorithm

is to find a hyperplane $\mathbf{w}\mathbf{x}+b$ that separates the instances with label -1 from those with label $+1$. Often, there exist multiple hyperplanes satisfying this requirement. Intuitively, the hyperplane that maximizes its distance to the nearest instances from each class would be the best choice. This distance is referred to as *margin*. Therefore the goal is to find the hyperplane defined by parameters $\mathbf{w}$ and b that separates the two classes and maximizes the margin.

Ideally, when the instances from the two classes are linearly separable, we have $y_i(\mathbf{w}\mathbf{x}_i+b) \geq 1$. That is, all training instances are correctly classified and located outside or on the margin. Note that the nearest instances from the two classes are located on the margin, which defines two parallel hyperplanes: $\mathbf{w}\mathbf{x}_1+b=-1$ and $\mathbf{w}\mathbf{x}_2+b=+1$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the nearest instances on the hyperplanes of the two different classes. The distance between the two hyperplanes, i.e., the margin can be derived at $\frac{\mathbf{w}}{||\mathbf{w}||_2}(\mathbf{x}_1-\mathbf{x}_2)=\frac{2}{||w||_2}$.

In summary, to find the hyperplane defined by w and b, the following optimization problem is formulated:

$$\min_{m,b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i}^{N}\xi_i, \tag{8.1}$$

subject to

$$\begin{cases} y_i(\mathbf{w}\mathbf{x}_i+b) \geq 1-\xi_i,\ i=1,...,N; \\ \xi_i \geq 0,\ i=1,...,N. \end{cases} \tag{8.2}$$

The first term in Eq. 8.1 corresponds to the maximization of the margin, and the second term introduces a slack variable ξ that allows certain classification errors in the training set in the case when the data set is not linearly separable. The parameter C is used to control the degree of tolerance for the errors.

Ranking SVM

Ranking SVM is a widely used technique for learning the structure of pairwise ordering between documents in Information Retrieval [106, 127]. Here, for simplicity, we follow the formulation in [127].

Given a set of documents $D=\{d_i\}_{i=1}^{m}$ and a set of queries $Q=\{q_k\}_{k=1}^{n}$, a set of linear ranking functions $f_{\mathbf{w}}(q)$ are defined so that the maximum number of the following inequalities is fulfilled:

$$\begin{gathered} \forall(d_i,d_j)\in r_1^*: \mathbf{w}\phi(q_1,d_i) > \mathbf{w}\phi(q_1,d_j), \\ \ldots \\ \forall(d_i,d_j)\in r_n^*: \mathbf{w}\phi(q_n,d_i) > \mathbf{w}\phi(q_n,d_j), \end{gathered}$$

where r^* is the optimal ranking of documents with respect to a query q in the training data where d_i is ranked higher than d_j, $\phi(q,d)$ is a feature vector that describes the

matching between a query q and a document d, and $\mathbf{w}$ is a weight vector that is to be learnt. Further, $i \neq j$ and the ordering is strict. For any weight vector $\mathbf{w}$, documents are ordered by their projection of feature vectors $\phi(q,d)$ onto $\mathbf{w}$. The optimization goal is therefore to find a weight vector $\mathbf{w}$ such that a minimum number of document pairs in the training set are disordered.

This optimization goal can be formulated as a binary classification problem on pairs of documents. Let $\mathbf{x}_{ijk} = \phi(q_k, d_i) - \phi(q_k, d_j)$, and the target value is defined as the ordering of the pair of documents given the query q_k:

$$y_{ijk} = \begin{cases} +1 & d_i \succ_r d_j \\ -1 & d_j \succ_r d_i, \end{cases} \tag{8.3}$$

where $\succ_r$ denotes an ordering of a pair of documents d_i and d_j under a ranking r.

Let a set of training examples consisting of a set of ranking lists $R = \{r_k\}_{k=1}^{n}$ with respect to a set of queries $Q = \{q_k\}_{k=1}^{n}$ and document collection be given. By adding the regularization term and the slack variable ξ to allow errors on training set, the SVM formulation of the classification problem is defined as the following constraint optimization problem:

$$\min_{\mathbf{w},\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum \xi_{ijk}, \tag{8.4}$$

subject to

$$\forall d_i \succ_r d_j, k \quad y_{ijk}\mathbf{w}\mathbf{x}_{ijk} \geq 1 - \xi_{ijk}, \tag{8.5}$$

$$\xi_{ijk} \geq 0. \tag{8.6}$$

Note that we only consider the situations $d_i \succ_r d_j$ in the constraint, (the situations that $d_j \succ_r d_i$ are implied by the given constraints), therefore y can be left out for simplicity:

$$\forall d_i \succ_r d_j, k \quad \mathbf{w}\mathbf{x}_{ijk} \geq 1 - \xi_{ijk} \tag{8.7}$$

We can see that Eq. 8.4 is very similar to Eq. 8.1. The only difference is the slack variable ξ. In binary SVM, a ξ_i is obtained with respect to a single instance x_i, and in ranking SVM, a ξ_{ijk} is obtained with respect to a document pair (d_i, d_j) for a query q_k. The learning algorithm aims to find a ranking which minimize the number of constraints being violated on the training set. For a given set of test documents and queries, the output of the learnt model can be used to sort the documents with respect to a given query, i.e., the projection of the feature vector $\phi(q,d)$ on the learnt weight vector $\mathbf{w}$:

$$RSV(q,d) = \mathbf{w}\phi(q,d). \tag{8.8}$$

8.3 Method

We consider the linking task as consisting of two sub-tasks, namely, the anchor detection task and target finding task. Following [178], we first solve the target finding task,

and then identify the links, i.e., anchor–target pairs. For both tasks, we use a machine learning based approach. Below, we formulate the anchor detection and target finding as a binary classification problem and as a ranking problem.

8.3.1 Learning problems for ALG

Now let us formulate the learning problems for the two sub tasks of ALG, anchor detection and target finding.

First, we look at the case of binary classification. For a given source text t, a set of ngrams NG^t is extracted and for each ngram $ng \in NG^t$ a set of candidate target pages C^{ng} are collected. For target finding, each pair (ng, c) is seen as an instance and a feature vector is constructed. A classifier is trained to classify pairs (ng, c) into {linked, not-linked}. The label "linked" indicates that c is a target for ng, and "not-linked" indicates that c is not a target for ng. For anchor detection, each of the ngrams ng is an instance and we classify it into one of the two classes: {anchor, non-anchor}.

Next, we formulate the ranking problems. For target finding, each ng can be seen as a query, and the set of candidate target pages are the corresponding documents that need to be ranked. The goal is to rank the candidate pages in C^{ng} in descending order from the most appropriate page to the least appropriate page as a target page for ng. For anchor detection, each source text t can be seen as a query, and the ngrams NG^t extracted from t can be see as the corresponding "documents." That is, all ngrams in NG^t are ranked in descending order from the most appropriate to least appropriate as an anchor text for t.

Below, we specify the features we use for training classification models and ranking models for the two tasks.

8.3.2 Features

We identify 6 types of feature for learning a preference relation between the candidate links. Table 8.2 specifies in which stage each type is used and Table 8.1 lists the features. Here, we discuss the motivations for using them and detail the formulation of some.

Ngram features

The ngram features suggest how likely a given ngram would be marked as an anchor text, without any other information such as its context in the source page, which includes its length, IDF score, the number of candidate targets associated with it, and its *ALR* (Anchor Likelihood Ratio) scores. IDF is calculated as

$$IDF(ng) = \log\left(\frac{|W|}{|\{d_i|1 \leq i \leq N, ng \in d_i, d_i \in W\}|}\right),$$

Ngram features	
Length(ng)	Number of words contained in the ngram
IDF(ng)	IDF score of the ngram
ALR(ng)	ALR score of the ngram, as detailed in Eq. 8.9
|Cand(ng)|	Number of candidate target pages associated with the ngram
Ngram - target features	
TitleMatch(ng, c)	Three values - 2: exact match; 1: partial match (i.e., either the title contains the ngram, or the ngram contains the title); 0: no match
RatioLink(ng, ct)	Link ratio of the ngram and the candidate target page, see Eq. 8.10
RatioAnchor(ng, c)	Anchor ratio of the ngram and the candidate target page, see Eq. 8.11
Ret_uni(ng, c)	Retrieval score with unigram model, i.e., BM25 with default parameter settings
Ret_dep(ng, c)	Retrieval scores with dependency model, i.e., Markov Random Field model as described in [174]
Rank_dep(ng, c)	Rank of the target page with the dependency retrieval model
Target features	
#Inlinks(c)	Number of in-links contained in the candidate target page
#Outlinks(c)	Number of out-links contained in the candidate target page
#Categories(c)	Number of Wikipedia categories associated with the candidate target page
Gen(c)	Generality of the candidate target page as described in [178]
Ngram - source features	
TFIDF(ng, t)	TFIDF score of the ngram in the source page
First(ng, t)	Position of first occurrence of the ngram in the source page, normalized by the length of the source page
Last(ng, t)	Position of last occurrence of the ngram in the source page, normalized by the length of the source page
Spread(ng, t)	Distance between first and last occurrence of the ngram in the source page, normalized by the length of the source page
Source-target features	
Sim(c, t)	Cosine similarity between the candidate target page and the source page
Ret_unigram(c, t)	Retrieval score using the title of the candidate target page as query against the source page; using BM25 as retrieval model
First stage scores	
score(ng, c)	Output of the ranker for the candidate target page given the ngram
rank(ng, c)	Rank of the candidate target page according to the learnt ranker

Table 8.1: Features used for learning the preference relation.

where d_i is a Wikipedia page containing this ngram. The *ALR* score can be interpreted as a model selection between two models using log likelihood ratio [166]. Assume we have two collections, the *anchor collection* A^W which contains all the anchor texts found in Wikipedia, and a *background collection* N^W, which contains all possible ngrams found in Wikipedia. Given an ngram *ng*, we compare the probability that it comes from A^W or N^W, if *ng* is randomly drawn from one of the collections.

Specifically, it is calculated as

$$ALR(ng) = \frac{|\{ng|ng \in A\}|}{|A^W|} \cdot \frac{|N^W|}{|\{ng|ng \in N\}|}, \tag{8.9}$$

A large *ALR* value indicates that the ngram is more likely to be an anchor text than a common word sequence from the background collection.

Ngram-target features

The ngram-target features describe how well an ngram and its corresponding candidate target page are related. On the assumption that each Wikipedia page is about a specific topic that is usually denoted by its title, the first feature we use is the match between an ngram and the candidate target page. The second type of feature in this category consists of indicators of how likely a given ngram *ng* and a candidate target page *c* are linked, which is expressed by the following two scores: *RatioLink* and *RatioAnchor*. The former is the ratio between the number of times *ng* and *c* are linked and the number of times *c* is being linked as a target page in the collection. The latter, i.e., *RatioAnchor*, is the ratio between the number of times *ng* and *c* are linked and the number of times *ng* is used as an anchor text in the collection:

$$RatioLink(ng,c) = \frac{|L_{ng,c}|}{|inlink(c)|} \tag{8.10}$$

$$RatioAnchor(ng,c) = \frac{|L_{ng,c}|}{|\{ng|ng \in A^W\}|} \tag{8.11}$$

Here, $|L_{ng,c}| = |\{l(a,d^*)|a = ng, d^* = c, d \in W\}|$ denotes the number of times that ngram *ng* and *c* are linked in Wikipedia, and $|inlink(c)|$ denotes the number of times that *c* is used as a target page and linked to from some anchor texts in Wikipedia.

Moreover, we adopt retrieval scores between the ngram and the candidate target pages as features (ngram as query), which is an obvious description of the relatedness of the two.

Target features

The target features are indicators of how likely a candidate target page alone would be linked with some anchor text in the collection. To this end we explore features such as counts of the inlinks and outlinks within the candidate target page, as well as the Wikipedia category information associated with it.

Ngram-source features

This type of feature describes the importance of the ngram within its context, i.e., source page. One would assume that an ngram being selected as an anchor text should

Learning Stage	Ngram	Ngram-target	Target	Ngram-source	Source-target	1st-stage
Candidate targets ranking		x	x		x	
Candidate links ranking	x	x	x	x	x	x

Table 8.2: Features and their corresponding application in different learning stages.

be somewhat important to the understanding of the whole source page as well as being content-wise related. Here, we use the TFIDF score of the ngram and its location within the source page as an indication of the importance of a ngram within a source page.

Source-target features

The source-target features describe the degree of relatedness between a source page and a candidate target page. One obvious feature is the similarity between the two pages. In addition, as a candidate target page itself is about a specific topic, we could measure how important this topic is, or in other words, how well this topic is being expressed in the source page. We measure it by using the title of the candidate target page as a query and calculating the retrieval score against the source page.

First stage score

As said, we first solve the target finding problem and then identify the anchor texts from a source text. Once target ranking has been completed (in the first stage), we get the ranking score and the rank of each candidate. In the second stage, we select the top X candidate targets to construct the candidate links with their corresponding ngrams, where the scores and ranks from the first stage are used as features.

8.4 Experiments

8.4.1 Training setup

We use two Wikipedia collections provided by INEX, namely the Wikipeida 2008 collection and the Wikipedia 2009 collection, to generate training data. We list some of the statistics of the two collections in Table 8.3.

For learning both the binary SVM and the RankingSVM, we randomly sample 500 pages from each of the Wikipedia collections for training and 100 pages for validation. For both SVMs we use the linear kernel and tune the regularization parameter C on the validation set. To learn the model for target finding, we use only the annotated anchor texts in Wikipedia and their corresponding candidate target pages as instances. The candidate target pages are collected using existing Wikipedia links. Over the training data, we assign a label $+1$ to the real target page of ng, and a label of -1 to the rest of the pages in C^{ng}. For training the anchor detectors we use all ngram-candidate target

Collection	Total pages	Total links
Wiki2008	659,388	17,018,711
Wiki2009	2,666,190	135,932,550

Table 8.3: Statistics of the two Wikipedia collections used in this chapter.

Run	**Description**
Wiki08_binary	Binary classification, trained on wiki08
Wiki08_rank	Ranking SVM, trained on wiki08
Heuristic	A heuristic run, combine the ALR and IDF for link ranking, but using rankingSVM for target ranking
Wiki09_binary	Binary classification, trained on wiki09
Wiki09_rank	Ranking SVM, trained on wiki09

Table 8.4: Description of 5 experimental runs.

pairs as instances. Here, we assign a label of $+1$ to the annotated anchor texts and for the rest of the ngrams in NG^t, we assign a label of -1.

We use the Weka [81] SVM implementation for training the binary classification SVM and SVMLight[1] for training the Ranking SVM. Note that since we will evaluate the resulting links using rank based evaluation metrics, we need to transform the result of binary classification to a ranked list. The Weka toolkit provides a confidence score for the predicted target value along with the classification results. We rank the resulting target pages and anchor texts in descending of their confidence scores of being a target page or an anchor text.

8.4.2 Experimental setup

With respect to the three research questions discussed in Section 8.1, we generate 5 runs as specified in Table 8.4. For the heuristic run we do not use a learning method for anchor text ranking; it only uses RankingSVM for target identification. For anchor detection, we filter the candidate links whose ALR score is less than 0.2, and rank the remaining ones with their IDF scores. The ALR and IDF scores are calculated over the Wikipedia 2009 collection. This run serves as a baseline for other machine learning based approaches. The heuristics used in this run, i.e., the *ALR* and *IDF* scores, however, are the features that are most close to human intuitions, where *ALR* represents how likely an ngram is involved in a link based on the observation of existing links and *IDF* represents the degree to which an ngram is uncommon.

[1] `http://svmlight.joachims.org/`

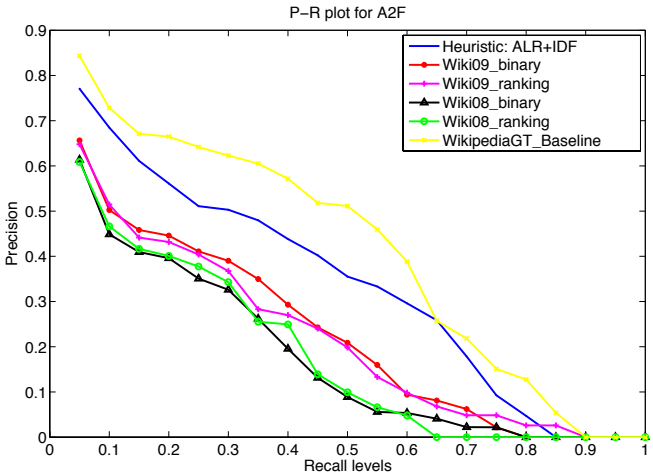


Figure 8.1: Precision-Recall plots for the 5 runs.

RunIDs	R@0.05	R@0.1	R@0.2	R@0.5
Wiki08_binary	0.61	0.44	0.39	0.09
Wiki08_rank	0.60	0.46	0.40	0.10
Heuristic	**0.77**	**0.68**	**0.56**	**0.35**
Wiki09_binary	0.65	0.50	0.44	0.21
Wiki09_rank	0.64	0.51	0.43	0.20
WikipediaGT	0.84	0.73	0.66	0.51

Table 8.5: Average precision at different recall levels, where R@X indicates the precision score at a recall of level X.

8.4.3 Evaluation

We use the INEX 2009 link-the-wiki topics as testing topics, which consist of 33 topics. Results reported here are actually submitted to the INEX2009 link-the-wiki track, and therefore all are manually accessed. As discussed in Section 2.3.3 on page 23, the INEX link-the-wiki track evaluates generated links at various levels. Here we focus on the anchor-to-file evaluation, as this is exactly the task of ALG addressed in this thesis. That is, a correct link consists of a correctly identified anchor and a correct target page for that anchor.

8.5 Results

Figure 8.1 shows the results of our 5 runs using a Precision-Recall plot. In addition, we list the precision scores at different recall levels in Table 8.5. The Wikipedia ground truth is included as a pseudo run and evaluated against the manual assessment. First of all, from Table 8.5, we see that runs trained on the Wikipedia 09 collection (i.e.,

Wiki09_binary, Wiki09_rank) outperform runs trained on the Wikipedia 08 collection (i.e., Wiki08_binary and Wiki08_rank). This suggests that a larger collection with more (existing) links provides better training materials. Also, we see that the runs based on binary classification methods (Wiki08_binary and Wiki09_binary) do not differ a lot from the learning to rank based runs (Wiki09_rank and Wiki09_rank). This may be due to the fact that the training examples from Wikipedia do not contain very strong ranking information, i.e., we only have two levels of judgement from the ground truth: "is a link" and "not a link."

Surprisingly, the heuristic run outperforms all sophisticated learning methods. This indicates that the two features, ALR and IDF, are very strong features that probably dominate the contribution to the learned models. However, the feature ALR depends very much on the statistics obtained from existing Wikipedia links and therefore it may be biased towards the "Wikipedia linking style." In other words, it captures well the pattern of Wikipedia links, but may not be effective if applied to a linking problem where the link structure is different from the Wikipedia links.

Finally, none of our runs outperforms the Wikipedia ground truth. This is no surprise, since the models are learned from the Wikipedia ground truth. In order to outperform the Wikipedia ground truth with a learning method, sufficiently many examples with manual labeling should be collected.

8.6 Conclusions

We have focused on exploring the effectiveness of applying machine learning approaches for the ALG task. We experimented with two types of learning approaches, namely classification and learning to rank. We evaluated the learning material for the task, where we used different sets of training data (based on different versions of Wikipedia). On top of that, we used a heuristic run to exam the impact of the features that are intuitively most effective.

We have found that the learning to rank based approach and the binary classification approach do not differ a lot. The more recent (2009) Wikipedia collection which is of larger size and has more links than the older (2008) collection, provides better training material for learning the models. None of the machine learning based approaches outperform the Wikipedia ground truth when evaluated with manual assessments, which suggests that in order to learn a better model for ALG in terms of agree with human assessments, more strict manual annotations[2] is necessary. In addition, the heuristic run outperforms all machine learning based runs, which suggests that the two features, ALR and IDF, are very strong features that capture the linking style of Wikipedia links very well. In the next chapter, we will discuss the impact of this type of feature in generating links to Wikipedia from data outside Wikipedia that has a different linking style.

[2]Note that Wikipedia links are manually annotated, but containing many trivial links that are actively rejected by human assessors.

Chapter 9

Automatic Link Generation for Radiology Reports

In the previous chapter, we empirically analyzed several factors that have an impact on automatic link generation with Wikipedia. In this chapter, we move from linking Wikipedia topics towards linking free texts to Wikipedia. The main research question we address in this chapter is as follows.

RQ7 Can state-of-the-art ALG systems that are, in principle, domain independent, be effectively applied to linking texts from a specific domain to Wikipedia? If not, can we improve the effectiveness of automatic link generation by considering domain specific properties of the data?

More specifically, we study a case where we use automatic link generation technology to annotate radiology reports with Wikipedia topics. Within this context, more specific research questions related to this main question are raised. See Section 9.1 below.

9.1 Introduction

We start by providing some background. Two trends are influencing the role of radiology in the care process. First, the services delivered by radiologists are becoming a commodity, that is, they can be delivered by any radiology party "without qualitative differentiation across [the] market".[1] This trend is caused by various technological advances and societal trends, such as teleradiology, picture archiving and communication systems, computer-aided diagnosis software, communication standards, and an increasing demand for cost effectiveness [21, 26, 169, 194]. A lively debate has ensued whether the commoditization of radiology is a desirable trend, and how it can be directed to safeguard the quality of care and the role of radiology in the care process [21, 22, 23, 68, 123, 136]. Second, the practice of radiology is influenced by the shift in medicine from a provider-centric model of care to a patient-centric model [117, 118].

[1] `http://en.wikipedia.org/wiki/Commodity`

This shift calls for improved and novel ways for radiologists to communicate with patients [169], which is especially challenging for radiology, as its practitioners typically have no direct contact with patients [193] (except for some subdisciplines, such as interventional radiology).

Both trends call for means to increase the value of radiology in the care chain, especially the value perceived by the patient, and preferably without increasing the radiologist's workload. The most important contribution of radiology to the care process are interpretations of radiology images that are communicated through narrative reports to, primarily, colleague clinicians [184, 195]. Various ways have been proposed to increase the economic value of reports, such as restructuring their contents [184], adding citations to the medical literature and embedding key images. These enhancements aim at increasing the value from the referring clinicians' point of view, but they do not necessarily serve the patients' interests [193].

In this chapter, we introduce a way to enhance radiology reports by adding links to Wikipedia. This scenario gives rise to the following tasks: given a radiology report, (i) mark the relevant medical phrases, and (ii) for each phrase marked, generate a link to a Wikipedia page that provides background information about the phrase. As defined in the context of Automatic Link Generation (ALG) (see Chapter 8), we refer to the first task as *anchor detection*, where the marked medical phrases are anchor texts, and refer to the second task as *target finding*. It is envisioned that these explanatory links, rendered as hyperlinks in the report, help the patient to understand the clinical vocabulary and the implications the report has for his or her medical situation. This will help to empower the patient in the care process and to reduce anxiety. Since the proposed system generates hyperlinks without human intervention, the annotation process does not put additional pressure on the radiologist's workload.

In principle, the techniques described in this chapter can be applied to any other medical knowledge source such as MedlinePlus,[2] produced and maintained by the National Library of Medicine. It can also be used to support other stakeholders such as referring physicians who may find it more useful to consult expert-level sources, for example Amirsys' STATdx encyclopedia.[3] Wikipedia is chosen mainly for the following reasons.

Quantity Wikipedia densely covers the medical domain. It contains medical lemmas from multiple medical thesauri and ontologies, for example International Statistical Classification of Diseases and Related Health Problems (ICD-9, ICD-10), Gray's Anatomy, etc.

Quality Although Wikipedia is written collaboratively by largely anonymous Internet volunteers, the quality of articles is guaranteed by the Wikipedia content criterion "verifiability," that is, readers should be able to verify the material in a Wikipedia

[2] `http://www.nlm.nih.gov/medlineplus`
[3] `http://www.amirsys.com`

page against a reliable source. In addition, errors in the content are often spotted quickly and corrected by collaborative editors [247].

Accessibility Wikipedia is a free online resource. All users can access its content without registering or creating an account. Moreover, the content of Wikipedia is usually written at a level understandable for patients, i.e., non-experts.

Maintenance The discussion tabs of a medical Wikipedia page generally contain a wealth of information that also documents changes to earlier version of the page.

Most of the studies in automatic link generation focus on solving a general problem, (e.g., developing an automatic link generation approach using Wikipedia as training material and applying it to any topic domain [175, 178]), or applying the link generation techniques in general domains that cover diverse sets of topics, (e.g., news, blogs, web, etc. [53, 126, 173]). Here, we focus on applying automatic link generation techniques to data from the radiology domain.[4]

Two features set radiology data apart from data from a general domain. On the one hand, medical phrases often have a regular syntactic structure. For example, they are often noun phrases with one or more modifiers (e.g., adjectives). Such regularity provides useful features for recognizing these medical phrases in the reports. On the other hand, in many cases, the presence of multiple modifiers as well as conjunctions within a single medical phrase results in a complex semantic structure. In other words, a complex topic structure in the context of this thesis. For example, the phrase "acute cerebral and cerebellar infarction" contains two topics "cerebellar infarction" and "acute cerebral infarction," where "cerebellar" and "cerebral" are synonyms. When linking this phrase to Wikipedia, one needs to identify the main topic it represents prior to searching for a target page in Wikipedia.

With the above mentioned properties in mind, we aim to develop an approach that takes into account the domain properties of the radiology reports, so that the effectiveness of link generation on radiology reports can be improved over those state-of-the-art systems that are developed for data from a general domain. Specifically, we seek answers to the following research questions:

RQ7a How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b How does our proposed approach compare to state-of-the-art approaches aimed at solving the automatic link generation problem in the general domain?

In addition, we notice that some medical phrases are more frequently seen than others. For example, "brain," as a relatively common topic in neuro-radiology, appears in almost all neuro-radiology reports, while "xanthogranulomas" only occurs in reports

[4] A radiology report is a semistructured document that in general consists of the following components: clinical information, e.g., symptoms, procedure of the radiology scan, findings from the scan, and impression, i.e., the opinion of the radiologist.

that discuss this specific medical condition. Here, we investigate how the frequencies of medical phrases are distributed over radiology reports and whether automatic link generation systems perform differently when dealing with medical phrases with different frequencies. Further, if there is a difference, how does the difference affect the overall performance of an automatic link generation system? Consequently, we formulate our third research question as

RQ7c What is the impact of anchor text frequency on the performance of automatic link generation systems?

We seek the answers to our research questions using empirical methods. A test collection is manually created for this investigation (see Section 9.3).

The contribution of the chapter can be summarized as follows. First, we propose an automatic link generation approach that aims at enhancing narrative radiology reports by automatically adding links from medical concepts in the reports to Wikipedia. This approach is shown to improve over two state-of-the-art link generation systems that have previously been developed to solve the automatic link generation problem in a general domain. Second, we conduct an in-depth analysis of the performance of both state-of-the-art systems and our proposed approach. The conclusions of our analysis provide useful hints for future work on this research topic.

The remainder of the chapter is organized as follows. Section 9.2 discusses related work in information extraction and mapping in the biomedical domain and applications in the radiology domain. In Section 9.3, we describe two state-of-the-art automatic link generation systems, which serve as baseline systems. In Section 9.4, we introduce our approach to automatically generate links for narrative radiology reports. Then in Section 9.5 we specify our experimental setup for evaluating our proposed approach. Section 9.6 compares the experimental results of our approach to that of the state-of-the-art systems, followed by a discussion on the factors that cause the difference in system performance in Section 9.7. In Section 9.8 we analyze the impact of anchor text frequency on the performance of automatic link generation systems. Section 9.9 concludes the chapter with answers to the research questions and a discussion of future directions for our work.

9.2 Information extraction and mapping for biomedical data

Natural language processing techniques have been widely applied in the biomedical domain to disclose information from clinical free-text documents. We highlight two tasks from medical natural language processing that are related to our work, namely, biomedical named entity recognition (NER) and concept mapping.

The NER task addresses identification of biomedical terminology, for example gene or protein names, from free text such as biomedical literature. This task is

very similar to the anchor text identification task we discuss in this chapter, which aims at identifying anatomy and diagnosis terms from radiology reports. The major biomedical NER methods fall into three categories [134]: dictionary-based approaches [5, 135, 220, 243, 244, 264], rule-based approaches [9, 70, 73, 74, 109, 240] and machine learning methods [45, 129, 148, 170, 224, 238, 273].

Compared to other types of approach, machine learning approaches have the advantage of being robust and flexible, as they generally generalize well beyond given vocabularies and easily adapt to new language styles. The machine learning techniques most commonly used in this area include the ones that are well-known for solving sequential labeling problems, such as Hidden Markov Models (HMM) [45, 273], Support Vector Machines (SVM) [129, 148, 238] and Conditional Random Fields (CRF) [170, 224]. Various types of feature have been explored, particularly syntactic features such as part-of-speech (POS) tags and orthographical features such as the combination of digits and letters. This is due to the fact that biomedical terminology, such as gene and protein names, often displays syntactic regularities as well as uncommon word spellings. In this chapter, we apply a CRF-based sequential labeling approach to our anchor text identification problem, as we have noticed that similar to gene and protein names, the annotated anchor texts in radiology reports display strong syntactic regularities.

The concept mapping task focuses on mapping names to concepts in a reference biomedical ontology, such as the Unified Medical Language System (UMLS)[5] and Medical Subject Headings (MeSH).[6] These ontologies attach one or more descriptions to each concept and interrelate concepts through a number of relation types. For a given biomedical name, the step of finding the most appropriate concept in the reference ontology resembles the target finding task of automatic link generation. Representative systems include MetaMap [10] and Peregrine [226]. In the development of these system, much effort has been devoted to resolving term variations and term ambiguity.

Some research programs have taken a more implicit viewpoint on concept mapping, in the sense that they do not map an explicit biomedical name to a concept from a reference ontology, but the entire body of text that contains the name. This type of mapping usually uses information retrieval techniques to rank the concepts from the reference data source in descending order of their relevance to the input text. For example, the EAGL system proposed by Ruch [206] assigns MeSH concepts to an input text using a retrieval system based on vector space models, and Trieschnigg et al. [242] use a retrieval system based on language models.

In the radiology domain, a number of information extraction systems have been developed that focus on narrative radiology reports. The Special Purpose Radiology Understanding System (SPRUS) [91] is one of the earlier systems of its kind that extracts and encodes findings and interpretations from chest radiology reports. The authors also experiment with syntactic extensions of SPRUS, reporting a 81% recognition rate in a small scale experiments (10 reports) [92]. The Medical Language Extraction and

[5] `http://www.nlm.nih.gov/research/umls/`

[6] `http://www.ncbi.nlm.nih.gov/mesh`

Encoding System (MedLee) [71] is a rule-based system designed to extract clinical information from clinical radiology reports and encode them in terms of a controlled vocabulary. It reports 70% recall and 87% precision scores on identifying four diseases from a set of 230 radiology reports. Recently, Soysal et al. [233] have proposed the Turkish Radiology Information Extraction System (TRIES) that extracts and converts clinical information from Turkish radiology reports based on manually crafted rules and a domain specific ontology. The authors report 93% recall and 98% precision scores on a corpus of abdominal radiology reports. The high performance, as stated by the authors, is mainly due to the effectiveness of the hand-crafted rules and the rich morphological structure of the Turkish language [233].

While all systems discussed above map terms to certain ontologies or thesauri, we aim to map the identified anchor texts to Wikipedia pages. Further, while our proposed system uses a machine learning based approach that does not require any external knowledge sources such as an ontology or hand crafted rules, it is flexible enough to be extended with this type of domain specific expert knowledge.

9.3 Two state-of-the-art automatic link generation systems

In this section, we discuss two state-of-the-art automatic link generation systems, namely Wikify! [175] and Wikipedia Miner [178]. We continue to use the notation as specified in Section 8.2.1 on page 137.

The procedure of automatically generating links from free text to Wikipedia can be divided into the following three components: (1) *anchor detection (AD)*; (2) *target candidate identification (TCI)*; and (3) *target detection (TD)*. Note that TCI and TD together can be seen as the target finding task. Here we decompose this task into two components because our proposed approach introduced in Section 9.4 has a major difference in the TCI component compared to the state-of-the-art systems introduced in this section. We define the following three functions corresponding to the three components discussed above: $AD(\cdot)$ detects a set of anchor texts A^t from the set of ngrams NG^t extracted from t; $TCI(\cdot)$ collects candidate target pages C^a from W with respect to an anchor text a, and $TD(\cdot)$ finds the target page d^* from C^a for a, i.e., identifies links $L^t = \{l_i(a,d^*)\}_{i=1}^{|L|}$.

9.3.1 Wikify!

The procedure by which the Wikify! system generates links from a source text t to a Wikipedia page can be summarized in the pseudo code illustrated in Algorithm 2. Below, we briefly describe the approaches the Wikify! system uses to implement the three functions $AD(\cdot)$, $TCI(\cdot)$ and $TD(\cdot)$.

Algorithm 2 Workflow of Wikify!

Input: NG^t
Output: L^t
$A^t = \emptyset$, $L^t = \emptyset$.
$A^t = AD(NG^t)$
for a in A^t **do**
  $C^a = TCI(a, W)$
  $d^* = TD(C^a, a)$
  $L^t \leftarrow L^t \cup \{l(a, d^*)\}$
end for
return L^t

Anchor detection For detecting anchor texts from NG^t, the Wikify! system ranks each ngram $ng \in NG^t$ according to a score and uses the top τ ranked ng's as anchor texts for t. Mihalcea and Csomai [175] have experimented with several scores, including TF.IDF, χ^2 and a *keyphraseness* score which turns out to be the most effective score among the three. The *keyphraseness* score is defined as follows.

$$keyphraseness = \frac{|A_{ng}|}{|D_{ng}|}, \tag{9.1}$$

where $|A_{ng}|$ is the number of Wikipedia pages where ng occurs as an anchor text, and $|D_{ng}|$ is the number of Wikipedia pages that mention the ngram ng.

Target candidate identification The Wikify! system collects C^a for a given a via existing Wikipedia links. That is, in Wikipedia, when an ngram is used as an anchor text in a source text, there exists a target Wikipedia page it links to, and this target page is selected as a candidate page. If the ngram has multiple interpretations, then different occurrences of this ngram may be linked to different target pages in Wikipedia, depending on the context of the occurrences.

Target detection To identify the target page d^* from C^a for a given a, Mihalcea and Csomai [175] have experimented with two approaches. The first one is a knowledge based approach, which selects the candidate target page that maximizes a score calculated using the Lesk [151] algorithm as the target page. The Lesk algorithm is used to calculate the word overlap between the candidate target page and the context where a occurs. The second approach uses a machine learning based approach. For each a, a classifier is trained to classify whether a candidate target page should be linked.

9.3.2 Wikipedia miner

The Wikipedia miner system implements the approaches proposed by Milne and Witten [178]. We summarize the workflow of the Wikipedia miner system using the

pseudo code in Algorithm 3.

Algorithm 3 Workflow of Wikipedia miner

Input: NG^t
Output: L^t
$A^t = \emptyset$, $L^t = \emptyset$, $Atmp = \emptyset$.
for ng in NG^t **do**
  $C^{ng} = TCI(ng, W)$
  $d^* = TD(C^{ng}, ng)$
  $Atmp \leftarrow Atmp \cup (ng, d^*)$
end for
$A^t = AD(Atmp)$
for a in A^t **do**
  $L^t \leftarrow L^t \cup \{l(a, d^*)\}$
end for
return L^t

Target candidate identification The Wikipedia miner system differs from the Wikify! system in that target candidate identification and target detection is performed over ngrams in stead of identified anchors. To collect target candidates, Wikipedia miner uses existing Wikipedia links. To improve efficiency, a threshold is used to filter out candidate pages that have very low chance of being linked to a given ngram ng based on the observations made from links among Wikipedia pages.

Target detection Wikipedia miner trains a classifier for target detection. One important feature is the relatedness of a candidate c to the context terms of an ngram ng. Specifically, a context term of an ngram ng is defined as an ngram that co-occurs with ng in t and is always linked to the same target page for all its occurrences within Wikipedia. A *relatedness* score is calculated to measure the semantic similarity of c and the target page of a context term by comparing their incoming and outgoing links. For more details of the features as well as the combination of features employed in the Wikipedia miner system, we refer to [178].

Anchor (link) detection Wikipedia miner does not have an explicit "anchor detection" phase, instead, anchor detection is achieved by detecting (a, d_a^*) pairs from all (ng, d_{ng}^*) pairs. Hence, the result of anchor detection is a set of links, since anchor texts are found together with their targets and each pair is classified as either "link" or "not a link." A classifier is trained over instances consisting of ngram-target pairs. Various features are used to train the classifier, including the keyphraseness score proposed in [175] and features reflecting the relatedness between source text and target page.

9.4 Method

We now proceed to introduce our approach. In general, the workflow of our own proposed system is the same as that of the Wikify! system, as illustrated in Algorithm 2. That is, it follows the following steps: anchor detection, target candidate identification and target detection.

9.4.1 Motivation

As discussed in Section 9.1, we have identified two properties of the anchor texts in radiology reports: the regularity of their syntactic structure and the complexity of their semantic structure.

In order to exploit the regularity of syntactic structure of medical phrases in the radiology reports, we treat the anchor detection problem as a sequential labeling problem. Sequential labeling is an effective approach in terminology recognition in various applications in the biomedical domain [45, 129, 148, 170, 224, 238, 273]. In addition, different from the two state-of-the-art systems, we learn the pattern of anchor texts from radiology data instead of Wikipedia. Intuitively, the anchor texts in Wikipedia are from a general domain and have a different syntactic structure from the medical anchor texts in the radiology data, therefore they may not provide effective training material for sequential labeling.

To cope with the complex semantic structure of the medical anchor texts, we propose a sub-anchor-based approach to retrieve candidate targets and to formulate features for target detection. By retrieving target candidates with respect to sub-anchors of an anchor text, we collect candidate pages that are potentially relevant to different topics contained in the anchor text. Then at the target detection phase, we aggregate features extracted at sub-anchor level to anchor-level. The feature of a single sub-anchor text is weighted by the importance of that sub-anchor, which is measured by its similarity to the original anchor text.

In the rest of this section, we first describe our approaches to the three components mentioned above. Then we summarize our approaches by giving an overview of our link generation system *LiRa*, which integrates these components.

9.4.2 Anchor detection

We define the sequential labeling task for anchor detection as follows. Given a text document, identify anchor texts by annotating each of the words in the text with one of the following labels: begin of anchor (BOA), in anchor (IA), end of anchor (EOA), outside anchor (OA), and single word anchor (SWA). SWA defines a single word anchor; BOA-(IA)-EOA defines an anchor with multiple words. Within this framework, we will use a conditional random fields (CRF) model [144], which has shown state-of-the-art performance in solving sequential labeling problems [170, 224].

Let $WS = w_1, \ldots, w_n$ be an observed word sequence of length n, and $SS = s_1, \ldots, s_n$ a sequence of states where s_i corresponds to the label assigned to the word w_i. Following Settles [224], we use linear-chain CRFs, which define the conditional probability of the state sequence given the observed word sequence as

$$p(SS|WS) = \frac{1}{Z(WS)} \exp \sum_{i}^{n} \sum_{k}^{m} \lambda_k f_k(s_{i-1}, s_i, w_i, i), \tag{9.2}$$

where $Z(WS)$ is a normalization factor over all state sequences, $f_k(\cdot)$ is a feature function and λ_k is a learnt weight for feature $f_k(\cdot)$. The feature function describes a feature corresponding to the position i of the input sequence, states at position i and $i-1$, and word at position i.

The goal of the learning procedure is to find the feature weights λ that maximize the log-likelihood of the training data:

$$LL = \sum_{i} \log p(s_i|w_i) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2}. \tag{9.3}$$

The second term in Eq. 9.3 is a spherical Gaussian weight prior [38] used to penalize the log-likelihood term to avoid over-fitting.

We use three simple features: the word itself, its part-of-speech (POS) tag and its syntactic chunk tag. We have also conducted preliminary experiments with several variations of these features, including orthographical features of the word, e.g., whether it contains digits, capitalization, as well as bigram and trigram features. However, the performance of these variations in anchor detection in terms of both precision and recall (see 9.5.3) are negligibly close to that of the three basic features. Therefore we focus on the three basic features.

9.4.3 Target candidate identification

Anchor decomposition

Given an identified anchor text a of length l, we decompose it into a set of all sub-sequences $S_a = \{s_i\}_{i=1}^{m}$, while keeping the original order of the words within the identified anchor text. For example, for the anchor text "white matter disease", we have a set of sub-sequences {"white", "matter", "disease", "white matter", "matter disease", "white disease", "white matter disease"}. We call those sub-sequences *sub-anchors*.

In addition, one feature of Wikipedia is that there exist redirect pages, which provide synonyms or morphological variations for a concept. For example, the concept "acoustic schwannoma" is redirected to "vestibular schwannoma." While decomposing an identified anchor text, we add those redirects to the set of sub-anchors, in order to reduce term mismatching and thus increase the recall of the annotated targets.

Candidate target retrieval

For each sub-anchor s, we retrieve a set of candidate target pages $C^s = \{c_i\}_{i=1}^n$, ranked in descending order of their *target probability*. Let $L_{s,c} = \{l(a,d^*)|a = s, d = c, d \in W\}$ denote all pairs of links found between s and c in Wikipedia links, that is, links between target page s and all occurrences of s as anchor texts. The target probability is calculated as

$$p(c_i|s) = \frac{|L_{s,c_i}|}{\sum_{j=1}^{n} |L_{s,c_j}|}. \tag{9.4}$$

We collect the top-K Wikipedia pages in terms of their target probability scores for each sub-anchor and use the union of all the collected pages from each sub-anchor as the candidate target pages for the anchor. When examining the occurrence of sub-anchor s in existing Wikipedia links, we consider partial matches of phrases. That is, if all terms in s appear ordered within a Wikipedia anchor text, it is considered to be an occurrence. In addition, if the title of a Wikipedia page matches s, we also include this page as a candidate target page.

9.4.4 Target detection

We use a machine learning based approach to identify the target page d^* for a given anchor text a. Specifically, we train a classifier over the *anchor-target – candidate* pairs (a,c), which are labeled as "link" or "non-link". We extract the following features to train the classifier: (i) title matching, (ii) target probability, and (iii) language model log-likelihood ratio. The first two features are calculated at the sub-anchor level, and the third feature is calculated for a candidate target page. Below, we explain each of the features.

Title matching

We consider the title matching scores for each of the sub-anchors. For a sub-anchor s of anchor a, and a candidate target page c, the title matching score is defined as follows:

$$tm(s,c) = f_{tm}(s,c)\frac{len(s)}{len(a)}, \tag{9.5}$$

where

$$f_{tm}(s,c) = \begin{cases} 1 & \text{if } s \text{ equals title of } c \\ 0 & \text{otherwise.} \end{cases}$$

and $len(\cdot)$ is number of words in a word sequence.

The title matching score reflects the degree of matching between the anchor text and the title of c. The longer the sub-anchor, the more similar the sub-anchor is to the original anchor text, and therefore we have a higher degree of matching between the anchor text and the title of c.

Target probability

As defined in Eq. 9.4, the target probability is the probability that a Wikipedia page will be selected as target page, given the anchor text. We calculate $p(c|s)$ for each sub-anchor s, which is in fact precomputed during the candidate retrieval procedure.

Since the target probability is calculated at the sub-anchor level, we need to aggregate those scores for the original anchor texts. Note that in the case of title matching, no explicit aggregation is needed, since for a given candidate target page, it can only match one of its sub-anchors. In the case of candidate target probability, we aggregate the features extracted from the sub-anchors into three features. For an anchor a and its sub-anchors S_a of a candidate target page c we define:

$$\begin{aligned}
\max_{\text{tp}}(c) &= \max_{s \in S} p(c|s); \\
\min_{\text{tp}}(c) &= \min_{s \in S} p(c|s); \\
\text{wsum}_{\text{tp}}(c) &= \sum_{s \in S_a} \frac{len(s)}{len(a)} p(c|s).
\end{aligned}$$

Language-model log-likelihood ratio (LLR)

The language-model log-likelihood ratio feature indicates to which extent a candidate target page is about radiology.

Language models are statistical models that capture the statistical regularities in generating a language [190]. Here we consider two language models. The first, θ_R, models the language used in the radiology reports, which we refer to as the *radiology model*, and the second, θ_W, models the language used in Wikipedia pages on topics in a general domain, which we refer to as *Wikipedia model*.

Each model defines a probability mechanism, which can be explained as follows. Assuming the two models sample terms from the radiology collection and the Wikipedia collection that follow a multinomial distribution, using a maximum likelihood estimation, the probability that a certain term t is selected given a collection can be estimated as the relative frequency of the term in the collection. Now, given a piece of text with n terms, $T = \{t_i\}_{i=1}^{n}$, the two models repeatedly sample n times, assuming independence between successive events. The probability that T is generated by the radiology language model can be defined as

$$p(t_1, t_2, \ldots, t_n|\theta_R) = \prod_{i=1}^{n} p(t_i|\theta_R), \tag{9.6}$$

while the probability that T is generated by the Wikipedia language model is

$$p(t_1, t_2, \ldots, t_n|\theta_W) = \prod_{i=1}^{n} p(t_i|\theta_W). \tag{9.7}$$

Given the above language models, we use the log-likelihood ratio (LLR) [166], a widely used model-comparison metric, to decide which model is more likely to have generated T:

$$\begin{aligned} LLR(T) &= \log\left(\frac{p(T|\theta_R)}{p(T|\theta_W)}\right) \\ &= \frac{\sum_{i=1}^{n} \log p(t_i|\theta_R)}{\sum_{i=1}^{n} \log p(t_i|\theta_W)}. \end{aligned} \tag{9.8}$$

To avoid zero probabilities, which come up if terms in T do not occur in the radiology reports or in Wikipedia, we use Laplacian smoothing [152]. That is, we assume that each word has been seen at least once.

The LLR score indicates which of the two models θ_R and θ_W is most likely to have generated T. A score larger than 0 indicates T is more likely to be generated by the radiology language model, hence more likely to be relevant to the anchor text identified from a radiology report.

In summary, we list the final features we use to train a classifier for identifying a target page from a set of candidate targets:

1 Title matching between a, c;
2 Maximum target probability $\max_{tp}$;
3 Minimum target probability $\min_{tp}$;
4 Weighted sum of target probability wsum_{tp};
5 Language model log-likelihood ratio of c.

9.4.5 LiRa: a system overview

In Figure 9.1, we show an overview of the architecture of the proposed system LiRa for automatically generating links from radiology reports to Wikipedia. When LiRa receives a radiology report, it first parses the report and extracts the features needed for sequential labeling. After sequential labeling, the identified anchor texts are passed to the next stage for target detection. For each anchor text, LiRa retrieves a set of candidate target pages, extracts features and submits to the trained classifier. The output of the classifier is aggregated and generates the final annotated reports.

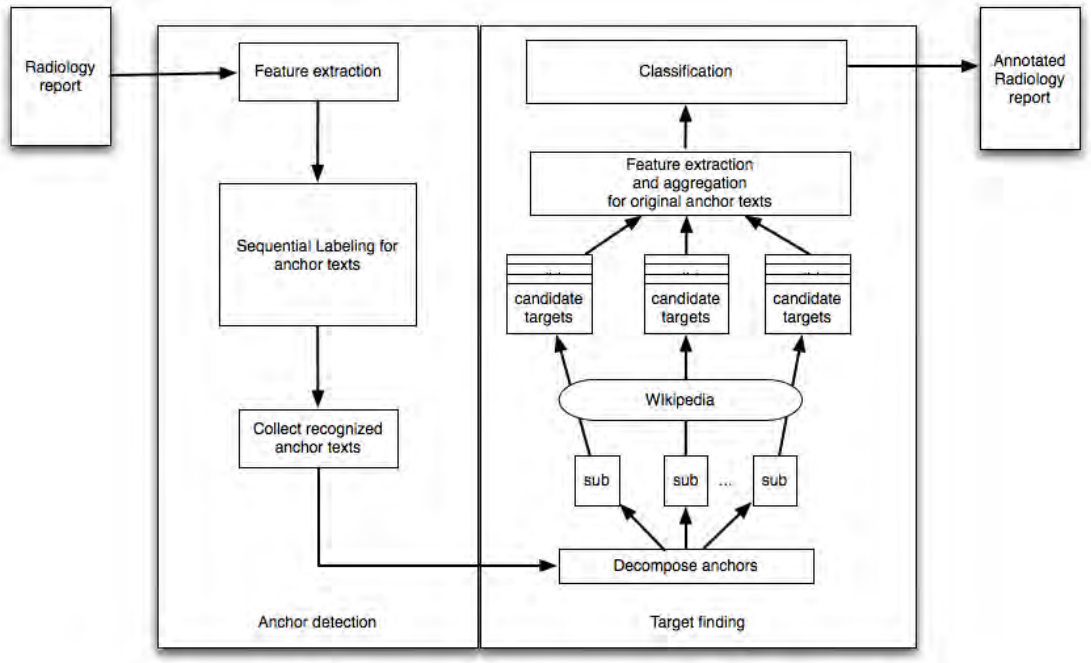


Figure 9.1: Architecture of the LiRa system.

9.5 Experiments

9.5.1 Research questions and experimental setup

Recall the research questions raised in Section 9.1:

RQ7a How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b How does our proposed approach compare to state-of-art approaches that are aimed at solving the automatic link generation problem in the general domain?

RQ7c What is the impact of anchor text frequency on the performance of automatic link generation systems?

To answer RQ7a, we offer our proposed automatic link generation approach as described in Section 9.4. We evaluate our approaches on the test collection that is developed on the purpose of evaluating automatic link generation for radiology reports. We describe the details of the collection in 9.5.2. We evaluate the systems on three aspects: (i) anchor text detection; (ii) target finding; and (iii) the overall performance of the system in generating the links.

For RQ7b, we run the two state-of-the-art link generation systems, namely Wikify! and Wikipedia miner, on the same test collection and compare their results against

the results of our proposed approach. We discuss the results and comparisons in Section 9.6. Note that in order to compare the performance of systems in target finding, we need to run the target finding components of each system on a same set of anchor texts. We include two sets of anchor texts for evaluation. First, the annotated anchor texts found in the ground truth can be used for this purpose. However, since we run the Wikipedia miner system as a black box (see Section 9.5.5), we do not have access to the intermediate result of target finding. Therefore this set can only be used to compare our system against the Wikify! system. The second anchor text set we consider is the anchor texts identified by the Wikify! system or by Wikipedia miner. That is, we run LiRa on the anchor texts identified by Wikify! (Wikipedia miner), and compare the target finding performance of LiRa against that of Wikify! (Wikipedia miner) on the same set of anchor texts.

On top of that, we provide two rounds of analysis. The first analysis, described in Section 9.7, further investigates the difference between the state-of-the-art systems and our proposed system in terms of their effectiveness of identifying anchors and linking them to the correct target pages. Particularly, we focus on the factors that make a system effective or non-effective.

The second analysis aims at answering RQ7c, where we compare the performance of the systems in anchor detection and target detection with respect to anchors with different frequencies. See Section 9.8.

9.5.2 Test collection

Our test collection is based on 860 deidentified neuroradiology reports, obtained from a US-based radiology institute. For the sake of the annotation process, the corpus was divided in three subsets; each subset was assigned to an annotator. Each annotator manually selected the anatomy and diagnosis phrases (i.e., the anchor texts in our experiments) in all reports assigned to him. The selections were stored as character ranges. Selections were allowed to overlap. For example, in the string "vestibular schwannomas," both "vestibular schwannomas" and "vestibular" were selected. The former is considered a diagnosis phrase, whereas the latter is considered an anatomy phrase.

For each selected phrase the annotator searched Wikipedia for the most appropriate page. All three annotators used Wikipedia's search engine. If a phrase did not have a directly matching Wikipedia page, a more general page was sought that reasonably covers the topic. If no such page was found, the phrase was assigned no Wikipedia page. Thus every phrase was assigned at most one Wikipedia page.

A home-grown annotation tool was used by all three annotators. Upon loading a new report, the tool selected phrases that were selected before by that annotator. The tool also suggested Wikipedia pages for phrases that were annotated before.

The three annotated subsets were merged and consolidated by a single annotator, thus yielding the test collection. In the consolidation phase, the following two properties were ensured:

- When a phrase is selected in one report, it is selected in all reports.
- Two occurrences of the same phrase, possibly in different reports, are assigned the same Wikipedia page, if any.

The second condition says that diagnosis and anatomy phrases are not ambiguous. In generally, this may be a strong assumption. For instance, in the medical domain the word "ventricle" is ambiguous as it may refer to a space in the heart as well as an area in the brain. In our corpus, however, it turned out to be a weak assumption. During the annotation process, no ambiguous phrases were encountered.

In total, 29,256 links, i.e., anchor text–target pairs, are extracted from the 860 reports, which can be resolved to 6,440 unique links. On average, each report contains 34 links.

As our target collection, we use the INEX 2009 Wikipedia collection [219].

9.5.3 Evaluation metrics

We use precision, recall and F-measure as our evaluation metrics. We evaluate the systems' performance on each radiology report, and show the overall performance which is averaged over all reports. Further, we use a paired t-test for significance testing. A $^{\blacktriangle}$ ($^{\blacktriangledown}$) indicates a significant increase (decrease) with p-value $<$0.01; and a $^{\triangle}$ ($^{\triangledown}$) indicates a significant increase (decrease) with p-value $<$0.05.

9.5.4 Preprocessing

We pre-process the Wikipedia collection as well as the radiology reports using Porter stemmer, in order to reduce the morphological variance of terms and phrases. When decomposing anchor texts to sub-anchors, we filter out word sequences that consist of function words only.

9.5.5 Parameter settings

In this section, we specify the parameter settings for each of the automatic link generation systems in our experiments.

Wikify!

We re-implement the Wikify! system as described in [175]. For anchor detection, following [175], we set the threshold τ to 6% of the length of the source text. Recall that τ is the threshold that selects the top X phrases ranked by the keyphraseness score as anchor texts.

Wikipedia miner

We use the online Wikipedia miner server[7] that is provided by the authors with default parameter settings. The server was accessed remotely and used as a black box.

LiRa

Anchor detection For anchor detection, we use the CRFsuite [185] implementation of CRFs with default parameter settings. For training and evaluating the anchor detection performance, we use 3-fold cross-validation.

As mentioned in Section 9.5.2, the annotations of the anchor texts can overlap. This poses a problem for the sequential labeling approach, as it allows us to assign only one label to each word. For example, in the case of "vestibular schwannomas," where both "vestibular schwannomas" and "vestibular" are annotated as anchor texts, we have to choose to assign either BOA-EOA or SWA to the word sequence when applying the sequential labeling procedure. In order to solve this problem, we construct the training set with two strategies. With the first strategy, for overlapping annotations, we choose the longer one, and with the second strategy, we choose the shorter one. We refer to the first strategy as *longest labeling (LL)*, and the second as *shortest labeling (SL)*. For example, in the case of "vestibular schwannomas," in the first setting, we use the label of BOA-EOA for "vestibular schwannomas" and ignore the anchor "vestibular", and in the second setting, we use the label of SWA for "vestibular" and ignore "vestibular schwannomas."

Target candidate identification At the target candidate identification stage, we rank Wikipedia pages in descending order of target probability scores and select the top K pages as candidate target pages. Heuristically, we set K to 10.

Target detection We calculate the LLR feature using the first 100 words of each candidate target page. There are two reasons why we select only the first 100 words. First, the first paragraph of a Wikipedia page is usually the summary of the content of that page, and therefore reflects the most important content of that page; the first 100 words is an approximation of the first paragraph of a Wikipedia page. Second, by using a constant number of words from each candidate target page, we eliminate the effect that the total number of words in the page has on the LLR score. This makes the LLR scores comparable across Wikipedia pages.

We experiment with three classifiers: Random Forest (RF) [27], Naive Bayes (NB) and SVM [47], using the Weka implementations [81]. After some preliminary experiments, we found out that RF always outperforms the other two classifiers in terms of both efficiency and effectiveness. Therefore, in the next section, we only focus on the results of RF. To train and evaluate the classifiers, 3-fold cross-validation is used as in the case of training the CRF model for anchor detection.

[7] `http://wikipedia-miner.sourceforge.net`

Along with the predicted labels, the classifiers also provide a score for prediction confidence. After classification, we execute a post-processing procedure. For anchor texts whose candidate target pages are all classified as "non-target," we select the candidate target that is predicted as "non-target" with the lowest prediction confidence as the target page. For anchor texts that have multiple candidate target pages classified as "target," we choose the one with the highest prediction confidence as the target.

9.6 Results

In this section, we show the effectiveness of our proposed approach to automatically generate links from radiology reports to Wikipedia as evaluated on our test collection.

Further, in order to answer RQ7b, *How does our proposed approach compare to state-of-the-art approaches that aimed at solving the automatic link generation problem in the general domain?*, we conduct a thorough comparison of the performance of our approach to that of the two state-of-the-art systems: Wikify! and Wikipedia miner.

9.6.1 Evaluation on anchor detection

System	precision	recall	F-measure
LiRa (LL)	**0.90**	0.80	**0.85**
LiRa (SL)	0.83$^{\blacktriangledown}$	**0.81**$^{\triangle}$	0.82$^{\blacktriangledown}$
Wikipedia miner	0.35$^{\blacktriangledown}$	0.36$^{\blacktriangledown}$	0.36$^{\blacktriangledown}$
Wikify	0.35$^{\blacktriangledown}$	0.16$^{\blacktriangledown}$	0.22$^{\blacktriangledown}$

Table 9.1: Results on anchor detection. LiRa (LL) represents the result of LiRa using longest labeling, and LiRa (SL) represents the result of LiRa using shortest labeling. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa(LL).

Table 9.1 lists the results of anchor detection for the three systems considered in our experiments. Here, two observations can be made. First, LiRa outperforms both Wikipedia miner and Wikify! in anchor detection in terms of all three evaluation metrics, i.e., precision, recall and F-measure. That is, the sequential labeling with CRFs approach trained on radiology data for anchor detection is more effective than the approaches employed by Wikify! and Wikipedia miner, where patterns of anchor texts are learnt from existing Wikipedia links. Second, comparing the two labeling strategies, i.e., LL versus SL, we notice that LL is more effective than SL in terms of precision and F-measure, while SL has a slightly better performance in terms of recall.

9.6.2 Evaluation on target finding

As discussed in Section 9.5.1, we compare the performance of the three systems in target finding using two sets of anchor texts. Table 9.2 shows the target finding performance of LiRa and Wikify! using annotated anchor texts found in the ground truth. In Table 9.3 we evaluate the performance of LiRa and Wikify! using anchor texts correctly identified by Wikify! and in Table 9.4 we compare the performance of LiRa and Wikipedia miner using the anchor texts correctly identified by Wikipedia miner.

From Table 9.2 and Table 9.3 we see that the target performance of LiRa is better than that of Wikify!. In both cases, i.e., using two different sets of anchor texts, the machine learning based target finding approach of Wikify! is more effective than the Lesk algorithm. Both approaches are less effective, however, than our proposed subanchor-based approach.

Further, from Table 9.4 we see that our proposed approach also outperforms the target finding approach of Wikipedia miner on the same set of anchor texts. The difference in performance between our approach and that of Wikipedia miner is less obvious than that between our approach and the Wikify! system.

System	precision	recall	F-measure
LiRa	**0.68**	**0.68**	**0.68**
Wikify (Lesk)	0.13$^{\blacktriangledown}$	0.13$^{\blacktriangledown}$	0.13$^{\blacktriangledown}$
Wikify (ML)	0.26$^{\blacktriangledown}$	0.26$^{\blacktriangledown}$	0.26$^{\blacktriangledown}$

Table 9.2: Comparing the performance of LiRa and Wikify! on target finding. The target finding algorithms are run on the annotated anchor texts found in the ground truth. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

System	precision	recall	F-measure
LiRa	**0.80**	**0.80**	**0.80**
Wikify (Lesk)	0.40$^{\blacktriangledown}$	0.40$^{\blacktriangledown}$	0.40$^{\blacktriangledown}$
Wikify (ML)	0.69$^{\blacktriangledown}$	0.69$^{\blacktriangledown}$	0.69$^{\blacktriangledown}$

Table 9.3: Comparing the performance of LiRa and Wikify! on target finding. The target finding algorithms are run on the anchor texts identified by Wikify!. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

9.6.3 Evaluation on overall system performance

Now we turn to the overall performance of our system in automatically generating links from radiology reports to Wikipedia, compared to the performance of the two state-of-the-art systems.

System	precision	recall	F-measure
LiRa	**0.89**	**0.89**	**0.89**
Wikipedia miner	0.84▼	0.84▼	0.84▼

Table 9.4: Comparing the performance of LiRa and Wikipedia miner on target finding. The target finding algorithms are run on the anchor texts identified by Wikipedia miner. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

In Table 9.5 we show the overall performance of the three systems, which is the final result of anchor detection and target finding. We see that LiRa outperforms the state-of-the-art systems in terms of overall performance, which is within our expectation, since we have already seen that for both anchor detection and target finding, LiRa has shown to be more effective than the other two systems. In addition, for LiRa, if we compare the performance of LL to SL in terms of overall performance, the limited difference in recall as shown in Table 9.1 in anchor detection has disappeared.

System	precision	recall	F-measure
LiRa(LL)	**0.65**	**0.58**	**0.61**
LiRa (SL)	0.60▼	**0.58**	0.59▼
Wikipedia miner	0.29▼	0.30▼	0.30▼
Wikify! (Lesk)	0.14▼	0.07▼	0.09▼
Wikify! (ML)	0.25▼	0.12▼	0.16▼

Table 9.5: Overall system performance. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa(LL).

In addition, in Table 9.6 we list the overall performance of Wikify! and LiRa using annotated anchor texts found in the ground truth as "recognized anchor texts." It can be seen as an oracle run of the two systems. That is, if all anchor texts can be correctly identified, we show the performance of the systems on linking these anchor texts to correct target pages in Wikipedia. We see that as in previous experiments, LiRa outperforms Wikify!. However, the performance of LiRa is far from perfect, leaving sufficient room for improvement.

System	precision	recall	F-measure
LiRa	**0.68**	**0.68**	**0.68**
Wikify! (Lesk)	0.13▼	0.13▼	0.13▼
Wikify! (ML)	0.25▼	0.25▼	0.25▼

Table 9.6: Overall system performance of Wikify! and LiRa in an oracle setting. Boldface indicates the best performance across systems. For significance testing, all runs are compared against LiRa.

9.6.4 Summary

In this section, we have provided a thorough comparison of our proposed approach to those approaches employed by state-of-the-art systems. With respect to the research question RQ7b, empirical results show that our approach is far more effective than the state-of-the-art approaches in terms of both anchor detection and target finding, and therefore overall performance as well.

In the next section, we further investigate factors that explain the performance difference between systems.

9.7 Discussion

From the description in Section 9.3 we can conclude that a common feature of the two state-of-the-art systems is that they both rely heavily on existing Wikipedia links. While the link structure in Wikipedia has been shown to provide useful training examples for automatic link generation systems in a general domain, it may not be as effective when used as training material for radiology data. As discussed in Section 9.4.1, medical phrases in radiology reports often have a complex semantic structure, for example containing multiple concepts as well as concepts with multiple modifiers. This is intuitively different from existing links in Wikipedia, where the semantic structure of an anchor text is usually less complicated. Or in other words, we expect that the pattern of annotated anchor texts in radiology reports are different from that of the anchor texts found in Wikipedia. Below, we investigate if the difference does indeed exist and whether it has an impact on system performance.

In total, we have 6,440 unique annotated anchor texts in our test collection. In Table 9.7 we list a set of statistics about the coverage of Wikipedia anchor texts over the annotated anchor texts found in our test collection. Let A^W be all the anchor texts found in Wikipedia. We evaluate the coverage on three aspects:

exact match the number of annotated anchor texts occurring in A^W;

partial match the number of annotated anchor texts occurring in A^W, including the cases when an annotated anchor text is a substring of a Wikipedia anchor;

sub exact match the number of annotated anchor texts containing at least one sub-anchor that occurs in A^W.

We see that very few ($<$20%) annotated anchor texts occur (fully or as as a sub string of the anchor texts) in A^W. However, over 80% of the annotated anchor texts do contain one or more concepts, i.e., sub-anchors, occurring in A^W.

Now let us look at what these statistics mean to the system performance. For anchor detection, both state-of-the-art systems rely heavily on the Wikipedia anchor texts. The keyphraseness score is used as the only score for identifying anchor texts in the Wikify! system, and used as an important feature for Wikipedia miner. However, from

Evaluation type	Occur. in WP links	coverage (%)
exact match	923	14.3
partial match	1,038	16.1
sub exact match	5,257	81.6

Table 9.7: The number of annotated anchor texts/sub-anchors in radiology reports covered by Wikipedia anchor texts.

Eq. 9.1, we can see that an anchor text only receives a non-zero score if it occurs in A^W. Given the low coverage of the annotated anchor texts in A^W, it is not surprising that the keyphraseness score is not effective, as around 85% of the annotated anchor texts would receive a 0 score. LiRa on the other hand, exploits the regularity of the syntactic structure of the annotated anchor texts in the radiology domain. The sequential labeling based approach captures this type of regularity, and is, therefore, effective for anchor detection.

For target detection, all three systems retrieve candidate target pages via Wikipedia links. The difference between the systems can be explained as follows. For Wikify!, candidate target pages are found with respect to an identified anchor text, and for Wikipedia miner, candidate target pages are found with respect to all possible ngrams extracted from a report, while for LiRa, candidate target pages are found with respect to the sub-anchors of an identified anchor text. It is obvious that the approach employed by Wikify! suffers from the same problem as in anchor detection: low coverage of Wikipedia anchor texts over annotated anchor texts in our test collection. LiRa solves this problem using its sub-anchor based approach to retrieve candidate target pages. From Table 9.7, we see that although not perfect, over 80% of the annotated anchor texts have the chance to retrieve their target pages. For Wikipedia miner, although a different strategy is employed, since all possible ngrams in a report are considered, the whole pool of candidate target pages at the report level cover a majority of the annotated target pages for that report. From Table 9.4 we see that this strategy achieves comparable results to our approach.

In summary, we conclude that the reasons why our proposed approach outperforms the state-of-the-art automatic link generation systems are as follows. The low performance of both state-of-the-art systems is mainly due to the complex semantic structure of the annotated anchor texts that are very different from the anchor texts found in Wikipedia. More specifically, the low coverage of Wikipedia anchor texts over the annotated anchor texts in the radiology reports is responsible for the low effectiveness of the two state-of-the-art systems. Our approach caters the complex semantic structure by employing a sub-anchor based approach to target finding and a sequential labeling based approach with syntactic features to anchor detection. Meanwhile the latter effectively exploits the syntactic regularity of medical phrases. Consequently, a much improved result is achieved by our proposed approach.

9.8 Further analysis

In this section, we turn to research question RQ7c: *What is the impact of anchor text frequency on the performance of automatic link generation systems?* More specifically, we investigate: (1) Does the performance of link generation systems show different patterns in recognizing and linking anchor texts with different frequencies of occurrence? (2) Further, if there is a difference, how does it influence the overall performance of the systems?

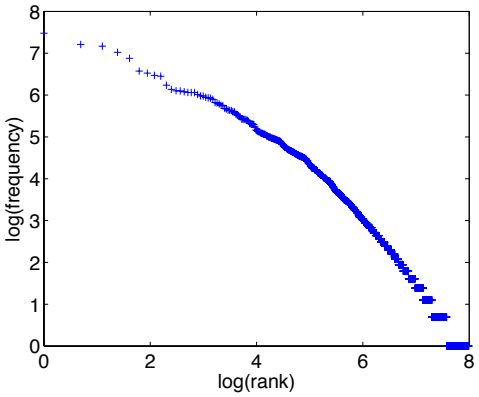
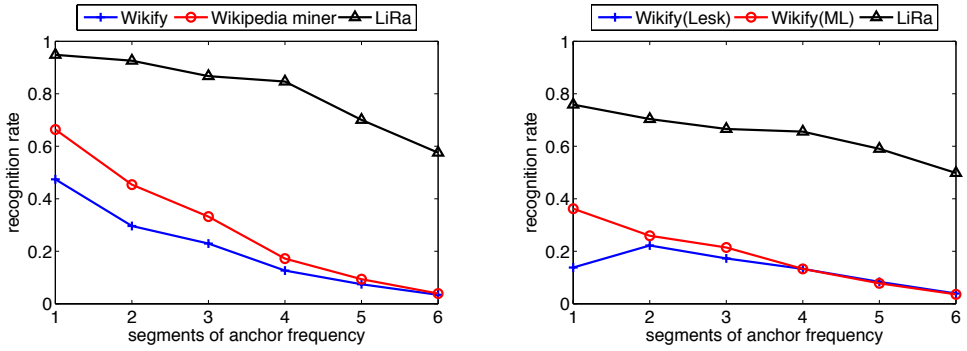


Figure 9.2: Distribution of anchor frequency. Anchors are ranked according to their frequency of occurrence in the radiology reports. The X-axis shows the logarithm of the ranks of anchors, and the Y-axis shows the logarithm of the frequency of the anchor at that rank.

Top 5	Bottom 5
mass	vestibular nerves
brain	virchow robins spaces
meningioma	warthins tumor
frontal	wegners granulomatosis
white matter	xanthogranulomas

Table 9.8: (Left) Five most frequent and (Right) five least frequent anchor texts found in the ground truth.

The motivation for the analysis conducted in this section is two-fold. In Figure 9.2, we rank the annotated anchor texts in decreasing order of their frequencies and plot their frequencies with respect to their ranks in the log scale. We see that the anchor text frequencies exhibit typical properties of Zipf's law [177]. The frequency of an anchor text is inversely proportional to its rank in the frequency table, which forms a distribution consisting of very few words with high frequencies and a long tail of anchor texts with low frequencies. Therefore if the frequency of an anchor text does



(a) (b)

Figure 9.3: Systems' performance differentiated by anchor text frequency. Anchors are ranked according to their frequency of occurrence in the radiology reports. The X-axes show the ranks of anchors, and the Y-axes show the systems' score on the r most frequent anchors, see Eq. 9.9. Figure 9.3(a) shows the anchor detection rate; Figure 9.3(b) shows the automatic link generation rate. See Table 9.9 for the segmentations of anchor texts based on their frequencies in the test collection

have an impact on the performance of a link generation system, it is important that a system can correctly recognize and link those rare anchors.

In addition, Table 9.8 shows the five most frequent and five least frequent anchor texts found in our test collection. Intuitively, frequent anchor texts are more likely to be common topics than infrequent anchor texts. By "common topics" we mean that the topic is frequently seen in a general domain and its meaning is more likely to be known to non-experts. For example, in Table 9.8 "brain" is more likely to be a common topic than "xanthogranulomas." We posit that common topics are more likely to occur in Wikipedia which makes it a relatively easy task for a link generation system, i.e., to identify it as an anchor text and find its target page.

In order to answer RQ7c, we divide the annotated anchor texts into different segments based on their frequencies, as listed in Table 9.9. We then evaluate the performance of the three systems in identifying and finding links for anchor texts in different segments. We evaluate the performance of a system on a segment *seg* using the following score:

$$score(seg) = \frac{tp_{seg}}{|seg|}, \tag{9.9}$$

where tp_{seg} is the number of anchor texts within the segment that are correctly recognized in the case of anchor detection, or whose target pages are correctly identified in the case of target finding.

Figure 9.3(a) shows the systems' performance at anchor detection and Figure 9.3(b) shows the systems' performance at target finding. Since we do not have access to the intermediate results of the Wikipedia miner system as discussed in Section 9.5.1, here we only show the performance of Wikify! and LiRa on target finding.

Segments	1	2	3	4	5	6
Freq. range	>100	51–100	11–50	6 –10	2–5	1
Num. anchors	116	108	527	482	1399	2149
Avg. freq.	271.1	70.1	20.7	6.5	2.6	1

Table 9.9: Segmentation of anchor texts based on their frequencies in the test collection.

For both anchor detection and target finding, we see a general trend that better performance is achieved on high frequency anchor texts compared to that on low frequency anchor texts. This observation holds for all systems, which suggests that in general, it may be an easier task for a link system to identify and to find targets for high frequency anchor texts than for low frequent anchor texts. In addition, we see that LiRa shows more robust performance compared to the other systems in that performance remains relatively high even on low frequency anchor texts.

In summary, with respect to research question RQ7c, we have the following answer. We find that anchor frequency has an impact on the performance of link generation systems in both anchor detection and target finding. Empirical results show that in general, link generation systems achieve better performance on high frequency anchor texts than on low frequency anchor texts. Further, since the distribution of anchor frequencies follows Zipf's law, it is important that a link generation system be effective on low frequency anchor texts, in order to achieve robust performance.

9.9 Conclusion

In this chapter, we have studied the problem of automatically generating links from radiology reports to Wikipedia. Two properties set our radiology data apart from data in a general domain, namely, the syntactic regularity and the semantic complexity of the anchor texts, i.e., medical phrases, found in radiology reports. Based on this observation, we proposed an automatic link generation approach for linking medical phrases from radiology reports to concepts in Wikipedia. Using a test collection developed in-house that consists of narrative radiology reports with manually annotated links to Wikipedia pages, we sought answers to three research questions:

RQ7a How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b How does our proposed approach compare to state-of-the-art approaches that aimed at solving the automatic link generation problem in the general domain?

RQ7c What is the impact of anchor text frequency on the performance of automatic link generation systems?

Our findings and our answers to the research questions can be summarized as follows.

To answer RQ7a, we use a sequential labeling based approach with syntactic features to anchor detection in order to exploit the syntactic regularity present among medical phrases. We then use a sub-anchor based approach to target finding, in order to resolve the complexity in the semantic structure of medical phrases. Our proposed approach has shown to be effective as evaluated on our test collection.

With respect to RQ7b, we find that our proposed approach outperforms two state-of-the-art systems in both anchor detection and target finding, and hence overall performance. Learning the linking patterns from the Wikipedia links, the two state-of-the-art systems failed to capture the domain specific properties of the radiology data, i.e., the syntactic regularity and semantic complexity of the anchor texts in the radiology reports.

Further, with respect to RQ7c, we find that automatic link generation systems tend to achieve better performance in recognizing and finding targets for annotated anchor texts with high frequencies compared to that achieved on anchor texts with low frequencies. Moreover, in order to achieve robust performance, it is important that a system is effective when dealing with low frequency anchor texts.

While our system has shown improved performance over existing automatic link generation systems in the radiology domain, several aspects of the automatic link generation techniques for radiology reports are worth further investigation. For example, in this chapter, we use a purely data-driven approach for both anchor text identification and target finding. An alternative route or extension of the route chosen in this chapter would consider symbolic knowledge representations that are widely available in the medical field, for instance in the form of ontologies. We believe that especially the task of finding a suitable generalization of an anchor text that does not have a matching page in Wikipedia can be achieved by following the hierarchical relationships in an ontology. This research agenda is closely connected to the recent MedlinePlus Connect[8] activity of the National Library of Medicine in which all SNOMED CT concepts are mapped to pages in Medline Plus.

[8] http://medlineplus.gov/connect

Conclusion to Part III

In the final part of the thesis, we addressed the research theme *relating topics in different representations*. More specifically, we focused on the task of Automatic Link Generation (ALG) with Wikipedia, which aims to identify significant terms or phrases in a piece of text, and for each term or phrase, generate a link to a Wikipedia page that provides background information for the term or phrase. Machine learning approaches using existing Wikipedia links as training data have shown satisfying performance on the related problem of (re)generating links between Wikipedia pages. In this part of the thesis, we evaluated "learning to link with Wikipedia" approaches in two different settings.

First, in Chapter 8 we evaluated the learning approaches in a setting where the task of ALG was formulated as a ranking problem, that is, for a given source text, links are identified and ranked according to their relevance to the source text, and for each anchor texts, target pages are ranked according to their relevance to the anchor text. Moreover, the resulting links were evaluated against manual assessments in stead of Wikipedia ground truth, i.e., existing Wikipedia links. Our main findings within this setting are as follows. (i) Linking models trained on a more recent Wikipedia collection (2009) which is of larger size and has more links achieve better performance compared to that achieved by models using an older Wikipedia collection (2008). (ii) Using a ranking based model (i.e., RankingSVM) does not outperform a binary classification based model (i.e., binary SVM), although the goal is to return a ranked list of links. (iii) When evaluating against human assessments, both Wikipedia ground truth and the links generated by models learnt from the Wikipedia ground truth are far from perfect.

Second, in Chapter 9 we turned to a second setting, where we aimed to investigate whether ALG systems that are trained domain independently can effectively link texts from a specific domain to Wikipedia. We conducted a case study in the radiology domain. We found that directly applying the domain independent ALG systems to the radiology data does not yield satisfying results. Further, our proposed ALG approach that considers domain specific properties of the radiology data has shown to effectively improve over the domain independent ALG systems.

Chapter 10

Conclusions

In this thesis, our aim has been to analyze and exploit topic structure in the context of Information Retrieval. Particularly, we have chosen to study the following aspects of topic structure, which we formulated as research themes: (i) topical coherence, (ii) diversity and the cluster hypothesis, and (iii) relating topics in different representations. In this chapter, we revisit the research questions we have posed in Chapter 1 with respect to these research themes and summarize our findings throughout the thesis. On top of that, we discuss a number of open issues and future directions.

10.1 Answers to the research questions

In Part I of the thesis we investigated topical coherence and its application in IR tasks. We started with the following research questions:

RQ1a. How do we measure the topical coherence of a set of documents?

RQ1b. Can the coherence score we propose effectively reflect the topical coherence of a set of documents?

We approached RQ1a from a document clustering point of view, where topics are represented by clusters of documents. Within this context, the topical coherence of a set of documents is associated with factors such as the number of topics found in the data set and the degree to which documents are focused on certain topic or topics. While determining the optimal number of clusters itself is a difficult problem, a more critical problem is that the two factors described above are both relative concepts that change when different topical granularity is considered.

Given the above thoughts, we proposed a coherence score in Chapter 3, which captures the topical coherence for a set of documents in an implicit way, that is, without explicitly modeling the topics and thus free of the assumption about the number of hypothesized topics. Specifically, the coherence score measures the topical coherence of a set of documents by comparing the distribution of the pairwise similarity scores

of the documents within this set to that of a set of documents randomly drawn from a background collection. The background collection serves as a reference point of topical granularity, which allows comparison of topical coherence between different document sets. We evaluated the coherence score on a toy data set as well as on simulated text data. In both cases, the coherence score was able to capture the relative degree of being focussed of documents on a single or multiple topics.

In Chapter 4 and 5 we evaluated the coherence score on two retrieval tasks, namely, the task of blog feed retrieval and the task of query performance prediction. The goal of the blog feed retrieval task is to identify blogs that show a central and recurring interest in a given topic. Keeping this goal in mind, in Chapter 4, we sought the answers to research questions RQ2a, RQ2b and RQ2c:

RQ2a. How do we measure topical consistency for a blog?

RQ2b. How can we use the coherence score in our blog retrieval process?

RQ2c. How does the size of a blog influence the estimation of the coherence score of the blog and how does this influence blog feed retrieval?

We used the coherence score as a measure of the topical consistency of the blog posts belonging to the same blog. We found that with a proper weighting scheme which controls the importance of topical consistency versus relevancy, incorporating the coherence score into a language modeling based retrieval model can significantly improve the performance of blog feed retrieval.

To predict query performance, we posited that in ad-hoc retrieval, queries with ambiguous terms tend to cause failures in finding relevant documents. Based on this assumption, we have proposed to use the coherence score of the set of documents associated with a query term found in the target collection as an indication of the level of its ambiguity. In this setting, we investigated the following research questions:

RQ3a. Can we use the coherence score to measure query ambiguity?

RQ3b. Can we use query ambiguity as measured by coherence-based scores to predict query performance in an ad-hoc retrieval setting?

We experimented with three ways to aggregate the term coherence scores for each query as query difficulty predictors. Empirical results have shown that our proposed predictors have significant positive correlation with the performance of the test queries in terms of AP on small collections.

In summary, we have proposed a coherence score to measure topical coherence of a set of documents and we were able to successfully apply this score to two retrieval tasks. While the coherence score has shown promising performance in both tasks, it is non-trivial to effectively apply it to a specific scenario.

In Part II of the thesis we studied the relation between topic structure and the relevancy of the documents retrieved with respect to a query. We started with the cluster hypothesis, according to which this relation can be interpreted as *relevant documents tend to be more similar to each other than to non-relevant documents* [105]. Given that the effectiveness of the cluster hypothesis has been validated through various cluster-based retrieval approaches in the context of ad-hoc retrieval, in this thesis, we re-visited this hypothesis in the context of result diversification, where the top ranked documents are expected to be both relevant and diverse. We asked the following research questions:

RQ4. How do we interpret the cluster hypothesis in the context of result diversification?

RQ5. Can query-specific clustering be used to improve the effectiveness of result diversification?

In order to answer RQ4, in Chapter 6, we empirically examined the validity of the cluster hypothesis with respect to a set of ambiguous or multi-faceted queries. Three specific research questions were formulated:

Q4a. Given a query that is ambiguous or multi-faceted, i.e., associated with several subtopics, do the relevant documents tend to be more similar to each other than to non-relevant documents? Particularly, do ambiguous or multi-faceted queries show different patterns in terms of inter-document similarities compared to specific or single-faceted queries?

Q4b. Do ambiguous queries show different patterns in terms of inter-document similarities compared to that of multi-faceted queries?

Q4c. Can we cluster the documents retrieved in response to an ambiguous or multi-faceted query in such a way that most relevant documents are contained in a small set of high quality clusters?

Our findings can be summarized as follows. First, our experimental results on the TREC2009 Web Track data suggest that with respect to ambiguous or multi-faceted queries, the cluster hypothesis is valid. Nevertheless, compared to specific or single-faceted queries, relevant documents associated with ambiguous or multi-faceted queries tend to have a less coherent topic structure. Second, we do not see a significant difference between ambiguous and multi-faceted queries in terms of inter-document similarities. By examining the queries in our test collection, we found that the distinction between ambiguous and multi-faceted queries is not clear-cut and that it is not sufficient to draw a conclusion based solely on the observations made on our 50 test queries. Third, we found that we can generate clustering structure desired by the cluster-based retrieval strategy for ambiguous or multi-faceted queries. In particular, LDA-based clustering is effective in gathering relevant documents into a small set of high quality clusters, without dominant clusters that contain most of the documents. Based on

these findings, we turn to RQ5 and investigate wether query specific clustering can be applied to improve the effectiveness of result diversification.

To answer RQ5, in Chapter 7 we proposed a result diversification approach based on query-specific clustering and cluster ranking, in which diversification is restricted to the documents belonging to a set of clusters that potentially contain a high percentage of relevant documents. Given the proposed approach to result diversification, we raised the following more specific research questions:

RQ5a. What is the impact of the proposed diversification framework on the effectiveness of existing result diversification methods? In other words, how much performance is gained by employing query-specific clustering and applying result diversification to documents contained in the top ranked clusters only?

RQ5b. What is the impact of the two main components, namely, the cluster ranker and the selection of number of top ranked clusters, on the overall performance of the proposed diversification framework?

RQ5c. Further, given that we use top ranked documents retrieved in response to a query for clustering as well as for diversification, how sensitive is the performance of the proposed framework to the number of documents being selected?

RQ5d. What conditions should clusters fulfill in order for diversification with cluster ranking to be effective?

We found that our proposed result diversification framework effectively improves the performance of several existing diversification methods. Further, the overall performance of our proposed approach is influenced by a number of factors, including the performance of the cluster ranker, the number of top ranked clusters being selected, and the length of the initial ranked list of documents. In our experiments, we used a query-likelihood based cluster ranker and determined the number of selected top ranked clusters with leave-one-out cross-validation. Although conceptually simple and not fully optimized, these choices lead to improved performance compared to result diversification without clustering and cluster ranking. Moreover, we examined the properties that clusters should have in order for our cluster-based diversification framework to be effective. Two properties are found to be important. First, most relevant documents should be contained in a small number of high quality clusters, while there should be no dominantly large clusters. Second, documents from these high quality clusters should have a diverse content.

Finally, in Part III of the thesis, we studied the problem of relating topics with different representations. More specifically, the goal is to relate the definition or description of a word or phrase in a knowledge base to the word in a piece of free text given its context. We study this problem in the scenario of Automatic Link Generation with Wikipedia

and formulated the following more specific research questions aiming at going beyond Wikipedia:

RQ6. While exploring Wikipedia's link structure for relating the two topical representations, what is the impact of the evaluation type, training collection and learning methods?

RQ7. Can the state-of-the-art ALG systems that are, in principle, domain independent, be effectively applied to linking texts from a specific domain to Wikipedia? If not, can we improve the effectiveness of automatic link generation by considering domain specific properties of the data?

For RQ6, in Chapter 8 we compared how machine learning based automatic link generation approaches behave given different training collections (different versions of Wikipedia) and learning approaches (classification versus learning to rank approaches). Three more specific research questions are raised with respect to RQ6.

RQ6a. When the ALG task is viewed as a ranking problem, is a learning to rank approach more effective than a binary classification approach?

RQ6b. Do different versions of the Wikipedia collection (with, potentially, differences in collection size, numbers of links, etc.) result in performance differences when used as training material?

RQ6c. Are the features used for learning the models effective? Are there single features whose contribution to the linking results is dominant?

Our results suggest that when evaluated with human assessments, a more recent Wikipedia version provides better training materials than an older version, where a more recent version typically contains more pages and a more dense link structure. Different learning approaches, however, do not show substantial differences, and a simple heuristic approach that combines the two strongest features effectively outperforms all learning based approaches. In addition, when evaluated against manual assessments, both our machine learning based approaches and the Wikipedia ground truth are far from perfect.

In response to RQ7, we conducted a case study in Chapter 9 using data from the radiology domain and seeking answers to the following more specific research questions.

RQ7a. How do we effectively annotate narrative radiology reports with background information from Wikipedia using automatic link generation techniques?

RQ7b. How does our proposed approach compare to state-of-the-art approaches aimed at solving the automatic link generation problem in the general domain?

RQ7c. What is the impact of anchor text frequency on the performance of automatic link generation systems?

We found that directly applying existing ALG systems trained on Wikipedia links to the radiology data does not yield satisfying results. Based on the observations made on the radiology data, we proposed our automatic link generation system which targets the radiology domain. We used a sequential labeling approach with syntactic features for anchor text identification in order to exploit the syntactic regularity present among medical phrases. Then we used a sub-anchor based approach to target finding, which was aimed at coping with the complex semantic structure of medical phrases. The proposed system effectively improves the performance in generating links for radiology data. Further, we found that in general, our ALG system tends to achieve better performance in recognizing and finding targets for annotated anchor texts with high frequencies than that achieved on anchor texts with low frequencies.

10.2 Future directions

In this thesis, we studied the impact and applications of topic structure in IR with a selection of three research themes. Within each research theme, there exist several perspectives that are not fully addressed or open issues that are worth further study.

We start with Part I of the thesis. The coherence measure we proposed in Chapter 3 can be improved in a number of aspects. First, the complexity of the current implementation of the coherence score is in the order of $O(n^2)$, where n is the number of documents within the set. For very large data sets, this complexity is undesirable. One possible solution is to explore sampling strategies such that the distribution of similarity scores can be approximated. Second, we have only experimented with cosine similarity as the similarity measure when calculating the coherence score, mainly because it is efficient to calculate and works effectively in practice. It is known that for clustering, using different types of similarity measures can result in different clustering structure for the same data set [254]. A recent experiment we have conducted also suggests that the distributions of similarity scores show different patterns when using different similarity measures, e.g., cosine similarity versus JS-divergence. The impact of this difference on the effectiveness of calculating and applying the coherence score remains to be investigated.

With respect to the application of the coherence score in blog feed retrieval, the following aspects need further investigation. First, while integrating coherence scores with the LM-based retrieval model, we have experimented with a number of weighting functions that aim to approximate the distribution of relevant information among a ranked list. More sophisticated weighting schemes should be considered, such as score regularization techniques [57, 58].

In Chapter 5, our experiments on using the coherence score for query performance prediction are preliminary. The experimental results by Hauff [90] have given rise to new research questions: *How do we use the coherence score for query performance prediction in large scale data collections?* The challenge is two fold. First, as said, the calculating coherence score with pair-wise similarity on large data collection itself

is computational demanding. Second, large collections such as the Web have their own features, such as link structure, spam, etc. Some of the features are problematic, such as spam, while others may be useful, such as the link structure of Web. It would be interesting to see the role of coherence among those features in predicting query performance. In addition, although proposed as a pre-retrieval predictor, coherence-based scores can also be used in a post-retrieval predictor.

In Part II of the thesis, the result diversification framework we proposed has shown its effectiveness and potential through extensive experiments. Several components of the proposed framework can be fine-tuned, such as the ranking of clusters, the parameter that determines the number of clusters being selected for diversification. Moreover, we illustrated the relation between relevance, diversity and the cluster hypothesis in an empirical way. Can we use formal models to describe their relations? Further, what is the impact of such a relation on tasks such as query performance prediction and score regularization?

In Part III, we studied a special case of the relatedness between topics: automatically linking the topics in implicit representations to their explicit representation. In this thesis, we focused on how the links should be established. A natural next step is to investigate how these links can be used in a "typical" retrieval setting. For example, recent work on conceptual language models [171] is closely related to this general goal, which aims to utilize the explicit representations of topics, referred to as *concepts*, to enhancing the estimation of query models.

Furthermore, the link generation approaches discussed in this thesis focus on one type of relation between topics, namely the explanatory relation. One interesting question here is that, can this type of approaches be generalized to identify other types of relations? A recent research focus on the *related entity finding* task as launched in TREC 2009 [14] may be a test bed where this type of exploration can be performed. In this task, a source entity is given and a target entity should be retrieved such that the target entity fulfills a specified relation with the source entity. While the entities can be seen as topics, represented either in an implicit way or an explicit way, finding a specific relation defines a similar task as finding links between two topics, with an additional constraint of the type of relation.

Appendix A

Hierarchical Agglomerative Clustering

Hierarchical clustering [128] can be agglomerative (bottom-up) or divisive (top-down). Here we specify the HAC algorithm applied in our study. Assume we have a document set $D = \{d_i\}_{i=1}^{N}$, a similarity metric $sim(d_1, d_2)$ that measures the similarity between two documents, and a linkage criterion $L(c_1, c_2)$ that measures the similarity between two clusters. The HAC algorithm is described as follows [120]:

1. Treat each document as a cluster. Compute a proximity matrix containing the (dis)similarity between each pairs of clusters using $L(\cdot)$.
2. Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.
3. If all documents are in one cluster, stop. Otherwise, go to step 2.

Different (dis)similarity metrics can be applied as $sim(\cdot)$ [120]. In this thesis we use the cosine similarity. Various linkage criteria were proposed in the literature [119, 133, 231, 253]. Here, we specify three commonly used linkage criteria that were used in this thesis, namely, single-linkage [231], complete-linkage [133] and Unweighted Pair Group Method with Arithmetic mean (UPGMA) [119].

The single linkage criterion measures the similarity between two clusters c_i, c_j by the maximum similarity between the documents from each of the clusters.

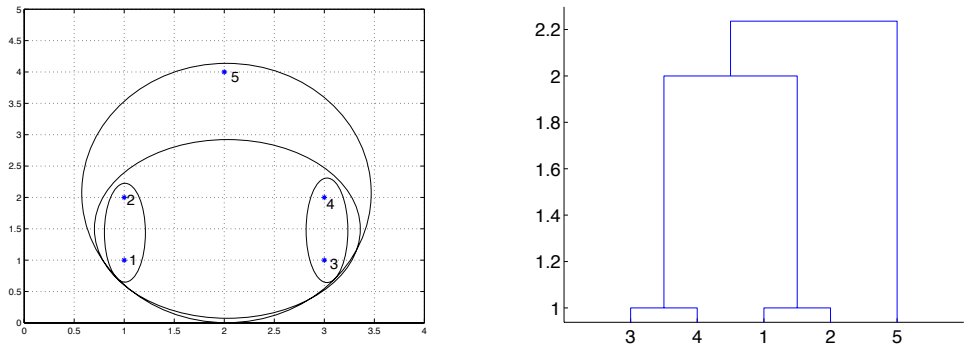
$$L_{\text{single-linkage}}(c_i, c_j) = \max \{sim(d_l, d_m) | d_l \in c_i, d_m \in c_j\}. \tag{A.1}$$

The complete linkage criterion measures the similarity between two clusters by the minimum similarity between the documents from each of the clusters.

$$L_{\text{complete-linkage}}(c_i, c_j) = \min \{sim(d_l, d_m) | d_l \in c_i, d_m \in c_j\}. \tag{A.2}$$

The UPGMA linkage criterion measures the similarity between two clusters by the average similarity between pairs of documents from each of the clusters.

$$L_{\text{UPGMA}}(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{d_l \in c_i, d_m \in c_j} sim(d_l, d_m). \tag{A.3}$$



(a) (b)

Figure A.1: Example of the dendrogram generated using a HAC algorithm using single linkage. Figure A.1(a) shows the data points to be clustered in a two dimensional space; and Figure A.1(b) shows the dendrogram generated on this data.

The output of the HAC algorithm is a nested hierarchy of graphs, referred to as *dendrogram*, which can be cut at a desired (dis)similarity level forming a partition (clustering) of documents. Figure A.1 shows an example of dendrogram generated using single linkage.

Bibliography

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09: Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009. Cited on pages 23, 29, 99, and 103.

[2] J. Allan. *Automatic hypertext construction*. PhD thesis, Cornell University, 1995. Cited on page 23.

[3] J. Allan and H. Raghavan. Using part-of-speech patterns to reduce query ambiguity. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 2002. Cited on page 82.

[4] I. S. Altingovde, E. Demir, F. Can, and O. Ulusoy. Incremental cluster-based retrieval using compressed cluster-skipping inverted files. *ACM Trans. Inf. Syst.*, 26:15:1–15:36, 2008. Cited on page 18.

[5] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997. Cited on page 151.

[6] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. on Info. Sys.*, 20:357–389, 2002. Cited on pages 15 and 73.

[7] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness and selective application of query expansion. In *ECIR'04: Proceedings of the 26th European Conference in Information Retrieval*, pages 127–137, 2004. Cited on page 69.

[8] E. Amitay, D. Carmel, A. Darlow, M. Herscovici, R. Lempel, A. Soffer, R. Kraft, and J. Zien. Juru at trec 2003 - topic distillation using query-sensitive tuning and cohesiveness filtering. In *TREC'03 Working Notes*, 2003. Cited on page 48.

[9] S. Ananiadou. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics*, volume 2, pages 1034–1038, 1994. Cited on page 151.

[10] A. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. In *Proceedings of AMIA Symposium*, pages 17–21, 2001. Cited on page 151.

[11] L. Azzopardi. Topic based language models for ad hoc information retrieval. In *In Proceedings of International Conference on Neural Networks and IEEE International Conference on Fuzzy Systems*, pages 3281–3286, 2004. Cited on page 19.

[12] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM Press, 2006. Cited on page 52.

[13] K. Balog, M. de Rijke, and W. Weerkamp. Bloggers as experts. In *SIGIR'08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 753–754. ACM, 2008. Cited on pages 46 and 52.

[14] K. Balog, A. P. de Vries, P.Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *TREC'09: Proceedings of the 18th Text REtrieval Conference*. NIST, 2009. Cited on page 181.

[15] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *ISTS'97: Proceedings of the Intelligent Scalable Text Summarization Workshop*, pages 10–17. ACL, 1997. Cited on pages 48 and 49.

[16] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *SIGIR'99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229. ACM, 1999. Cited on page 15.

[17] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. Cited on page 18.

[18] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. Cited on pages 3, 5, 17, and 87.

[19] W. Bluestein. *Hypertext versions of journal articles: Computer aided linking and realistic human evaluation*. PhD thesis, University of Western Ontario, 1999. Cited on page 23.

[20] H.-H. Bock. On some significance tests in cluster analysis. *Journal of classification*, 2(1):77–108, 1985. Cited on page 5.

[21] G. W. L. Boland. The impact of teleradiology in the United States over the last decade: driving consolidation and commoditization of radiologists and radiology services. *Clin Radiol*, 64:457–460, 2009. Cited on page 147.

[22] G. W. L. Boland. Teleradiology for auction: the radiologist commoditized and how to prevent it. *J Am Coll Radiol*, 6:137–138, 2009. Cited on page 147.

[23] J. P. Borgstede. Radiology: commodity or specialty? *Radiology*, 247:613–616, 2008. Cited on page 147.

[24] G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for experimenters*. John Wiley & Sons, 1978. Cited on page 30.

[25] B. R. Boyce. Beyond topicality: A two stage view of relevance and the retrieval process. *Information Processing & Management*, 18(3):105–109, 1982. Cited on pages 22 and 82.

[26] W. G. Bradley. Off-site teleradiology: the pros. *Radiology*, 248:337–341, 2008. Cited on page 147.

[27] L. Breiman. Random forest. *Machine Learning*, 45(1):5–32, 2001. Cited on page 163.

[28] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR'00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40. ACM, 2000. Cited on page 28.

[29] F. Can. Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.*, 11: 143–164, 1993. Cited on page 18.

[30] F. Can and E. A. Ozkarahan. Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.*, 15:483–517, 1990. Cited on page 18.

[31] F. Can, I. S. Altingövde, and E. Demir. Efficiency and effectiveness of query processing in cluster-based retrieval. *Inf. Syst.*, 29:697–717, 2004. Cited on page 18.

[32] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998. Cited on pages 22, 99, and 102.

[33] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Morgan and Claypool Publishers, 2010. Cited on page 69.

[34] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41:17:1–17:38, 2009. Cited on page 18.

[35] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *CIKM '09: Proceedings of the 18th ACM conference on Information and knowledge*

management, pages 1287–1296. ACM, 2009. Cited on pages 22, 23, 88, 99, 103, and 115.

[36] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM'09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM, 2009. Cited on page 28.

[37] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436. ACM, 2006. Cited on pages 22 and 99.

[38] S. F. Chen and R. Rosenfeld. A gaussian prior for smoothing maximum entropy models. Technical report, Carnegie Mellon University, 1999. Cited on page 156.

[39] C. Clarke. Overview of the TREC 2004 terabyte track. In *TREC'04: Proceedings of the 13th Text REtrieval Conference*, 2004. Cited on page 74.

[40] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 web track. In *TREC'09: Proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*. NIST, 2010. Cited on pages 20, 29, 83, 99, 101, 109, and 110.

[41] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. Mackinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008. Cited on pages 28, 29, and 109.

[42] C. L. A. Clarke, N. Craswell, I. Soboroff, and G. V. Cormack. Preliminary overview of the trec 2010 web track. In *TREC'10 working notes*, 2010. Cited on pages 29 and 87.

[43] C. W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Aslib cranfield research project, College of Aeronautics, 1962. Cited on pages 1 and 25.

[44] C. W. Cleverdon, J. Mills, and E. Keen. Factors determining the performance of indexing systems. Aslib cranfield research project, College of Aeronautics, Cranfield, UK, 1966. Cited on pages 1, 12, 25, and 26.

[45] N. Collier, C. Nobata, and J.-i. Tsujii. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the 18th conference on Computational linguistics*, volume 1, pages 201–207, 2000. Cited on pages 151 and 155.

[46] W. S. Cooper. A definition of relevance for information retrieval. *Pergamon Press*, 7:19–37, 1971. Cited on page 1.

[47] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. Cited on pages 136 and 163.

[48] W. B. Croft. *Organizing and searching large files of document descriptions*. PhD thesis, Churchill College, University of Cambridge, 1978. Cited on page 19.

[49] W. B. Croft. A file organization for cluster-based retrieval. In *SIGIR'78: Proceedings of the 1st annual international ACM SIGIR conference on Information storage and retrieval*, pages 65–82. ACM, 1978. Cited on page 18.

[50] W. B. Croft. A model of cluster searching based on classification. *Information Systems*, 5: 189–195, 1980. Cited on pages 6, 18, and 19.

[51] S. Cronen-Townsend and W. B. Croft. Quantifying query ambiguity. In *HLT'02: Proceedings of the second international conference on Human Language Technology Research*, pages 104–109, 2002. Cited on pages 69, 70, 72, 73, and 74.

[52] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR'02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2002. Cited on pages 69 and 70.

[53] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP-CoNLL'07:Conference on Empirical Methods in Natural Language Processing Conference on Computational Natural Language Learning Joint Meeting following ACL 2007*, pages 708–716, 2007. Cited on pages 23 and 149.

[54] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR'92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM, 1992. Cited on page 19.

[55] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990. Cited on pages 5 and 17.

[56] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 2006. Cited on page 136.

[57] F. Diaz. Regularizing ad hoc retrieval scores. In *CIKM'05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 672–679. ACM, 2005. Cited on page 180.

[58] F. Diaz. Regularizing query-based retrieval scores. *Information Retrieval*, 10:531–562, 2007. Cited on page 180.

[59] R. Dubes and A. K. Jain. Validity studies in clustering methodologies. *Pattern Recognition*, 11 (4):235–254, 1979. Cited on page 42.

[60] A. El-Hamdouchi and P. Willett. Comparison of hierarchic agglomerative clustering methods for document retrieval. *Comput. J.*, 32:220–227, 1989. Cited on page 19.

[61] A. El-Hamdouchi and P. Willett. Techniques for the measurement of clustering tendency in document retrieval systems. *J. Inf. Sci.*, 13:361–365, 1987. Cited on page 18.

[62] D. Ellis, J. Furner, and P. Willett. On the creation of hypertext links in full-text documents: Measurement of retrieval effectiveness. *Journal of the American Society for Information Science*, 47(4):287–300, 1996. Cited on page 23.

[63] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and feedback models for blog distillation. In *TREC'07 Working Notes*. NIST, 2007. Cited on page 21.

[64] B. J. Ernsting, W. Weerkamp, and M. de Rijke. The University of Amsterdam at the TREC 2007 Blog Track. In *TREC'07 Working Notes*. NIST, 2007. Cited on page 21.

[65] B. S. Everitt. Unresolved problems in cluster analysis. *Biometrics*, 35(1):169–181, 1979. Cited on page 5.

[66] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998. Cited on page 50.

[67] S. Fissaha Adafre and M. de Rijke. Discovering missing links in Wikipedia. In *Proceedings of LinkKDD-2005 Workshop*, 2005. Cited on page 23.

[68] R. Fitzgerald. Commentary on the impact of teleradiology in the United States over the last decade: driving consolidation and commoditization of radiologists and radiology services. *Clinical Radiology*, 64:461–462, 2009. Cited on page 147.

[69] E. A. Fox. *Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types*. PhD thesis, Cornell University, 1983. Cited on page 13.

[70] K. Franzen, G. Eriksson, and F. Olsson. Protein names and how to find them. *International Journal of Medical Informatics*, 67(1):49–61, 2002. Cited on page 151.

[71] C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson. A general natrual language text processor for clinical radiology. *Journal of the American Medical Informatics Assoiation*, 1(2):161–174, 1994. Cited on page 152.

[72] K. Fujimura, H. Toda, T. Inoue, N. Hiroshima, R. Kataoka, and M. Sugizaki. Blogranger—a multi-faceted blog search engine. In *WWW'06: Proceedings of the 15th World Wide Web Conference*, 2006. Cited on page 46.

[73] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of Pacific Symposium on Biocomputations*, pages 707–718, 1998. Cited on page 151.

[74] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: The pasta system. *Bioinformatics*, 19(1):135–143, 2003. Cited

on page 151.

[75] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. Cited on page 88.

[76] H. P. Giger. Concept based retrieval in classical ir systems. In *SIGIR'88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–289. ACM, 1988. Cited on page 12.

[77] W. Goffman. A searching procedure for information retrieval. *Information Storage and Retrieval*, 2(2):73–78, 1964. Cited on pages 22 and 82.

[78] S. Green. Building newspaper links in newspaper articles using semantic similarity. In *Proceedings of the Natural Language and Data Bases Conference*, pages 178–190, 1997. Cited on page 23.

[79] A. Griffiths, H. Luckhurst, and P. Willett. Using inter-document similarity information in document retrieval systems. *Journal of the American Society for Information Science*, 37:3–11, 1986. Cited on pages 18 and 19.

[80] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004. Cited on pages 17 and 88.

[81] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, 2009. Cited on pages 144 and 163.

[82] M. A. K. Halliday and R. Hasan. *Cohesion in English (English Language)*. Longman Pub Group, 1976. Cited on page 48.

[83] J. Han and M. Kamber. *Data mining*. Morgan Kaufmann Publisher, 2001. Cited on page 19.

[84] D. Harman, editor. *Proceedings of the Third Text Retrieval Conference TREC-3*. NIST, 1995. Cited on page 83.

[85] D. Harman and C. Buckley. The NRRC reliable information access (RIA) workshop. In *SIGIR'04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 528–529, 2004. Cited on pages 69 and 70.

[86] S. P. Harter. The cranfield II relevance assessments: A critical evaluation. *Library Quarterly*, 41 (3):229–243, 1971. Cited on page 1.

[87] S. P. Harter. A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. *J. Am. Soc. Inf. Sci.*, 26(5):280–289, 1975. Cited on page 15.

[88] S. P. Harter. Psychological relevance and information science. *Journal of American Society for Information Science and Technology*, 43:602–615, 1992. Cited on page 2.

[89] J. A. Hartigan. Statistical theory in clustering. *Journal of classification*, 2(1):63–76, 1985. Cited on page 5.

[90] C. Hauff. *Predicting the effectiveness of queries and Retrieval Systems*. PhD thesis, University of Twente, 2010. Cited on pages 69, 73, and 180.

[91] P. Haug, D. Ranum, and P. Frederick. Computerized extraction of coded findings from free-text radiologic reports. *Radiology*, 174(2):543–548, 1990. Cited on page 151.

[92] P. Haug, S. Koehler, L. M. Lau, P. Wang, R. Rocha, and S. Huff. A natural language understanding system combining syntactic and semantic techniques. In *Proceedings of Annual Symposium on Computer Application in Medical Care*, pages 247–151, 1994. Cited on page 151.

[93] B. He and I. Ounis. Query performance prediction. *Inf. Syst.*, 31(7):585–594, 2006. Cited on page 69.

[94] B. He, C. Macdonald, J. He, and I. Ounis. An effective statistical approach to blog post opinion retrieval. In *CIKM'08: Proceeding of the 17th ACM conference on Information and knowledge management*, 2008. Cited on page 9.

[95] J. He. Link detection with Wikipedia. In *INEX'08*, pages 366–373, 2008. Cited on page 9.

[96] J. He. Topic structure for information retrieval. In *SIGIR'09: Proceedings of the 32nd Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, page 850. ACM, 2009. Cited on page 9.

[97] J. He and M. de Rijke. An exploration of learning to link with wikipedia: Features, methods and training collection. In *INEX'09*, pages 324–330, 2009. Cited on page 9.

[98] J. He and M. de Rijke. A ranking approach to target detection for automatic link generation. In *SIGIR'10: Proceedings of the 33rd Annual International ACM SIGIR Conference*, pages 831–832. ACM, 2010. Cited on pages 9 and 24.

[99] J. He, M. Larson, and M. de Rijke. On the topical structure of the relevance feedback set. In *WIR'08*, 2008. Cited on page 9.

[100] J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *ECIR'08: Proceedings of the 30th European Conference in Information Retrieval*, 2008. Cited on pages 9, 37, 69, and 73.

[101] J. He, W. Weerkamp, M. Larson, and M. de Rijke. Blogger, stick to your story: modeling topical noise in blogs with coherence measures. In *AND '08: Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 39–46. ACM, 2008. Cited on pages 9 and 37.

[102] J. He, W. Weerkamp, M. Larson, and M. de Rijke. An effective coherence measure to determine topical consistency in user generated content. *International Journal on Document Analysis and Recognition*, 12(3):185–203, 2009. Cited on pages 9 and 37.

[103] J. He, M. Sevenster, and M. de Rijke. Automatically generating explanatory links for narrative radiology reports (submitted). *Journal of Biomedical Informatics*, 2010. Cited on page 9.

[104] J. He, E. Meij, and M. de Rijke. Result diversification based on query specific cluster ranking. *Journal of American Society for Information Science and Technology*, 62(3):550–571, 2011. Cited on pages 9 and 37.

[105] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84. ACM, 1996. Cited on pages 6, 18, 19, 37, 82, 83, 86, 90, 100, and 177.

[106] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*, chapter 7, pages 115–132. MIT Press, Cambridge, MA, 2000. Cited on page 138.

[107] W. Hersh. Relevance and retrieval evaluation: perspectives from medicine. *Journal of American Society for Information Science and Technology*, 45(3):201–206, 1994. Cited on page 2.

[108] D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *ECDL'98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 569–584. Springer-Verlag, 1998. Cited on page 15.

[109] J. Hobbs. Information extraction from biomedical text. *Journal of Biomedical Informatics*, 35 (4):260–264, 2002. Cited on page 151.

[110] T. Hofmann. Probabilistic latent semantic analysis. In *UAI'99: Proceedings of Uncertainty in Artificial Intelligence*, pages 289–296, 1999. Cited on pages 5 and 17.

[111] D. W. C. Huang, Y. Xu, A. Trotman, and S. Geva. *Overview of INEX 2007 Link the Wiki Track*, pages 373–387. Springer-Verlag, 2008. Cited on page 24.

[112] D. W. C. Huang, S. Geva, and A. Trotman. *Overview of the INEX 2008 Link the Wiki Track*, pages 314–325. Springer-Verlag, 2009. Cited on page 24.

[113] D. W. C. Huang, A. Trotman, and S. Geva. The importance of manual assessment in link discovery. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 698–699. ACM, 2009. Cited on pages 24 and 136.

[114] D. W. C. Huang, S. Geva, and A. Trotman. Overview of the INEX 2009 link the wiki track. In *INEX'09*, pages 312–323. Springer-Verlag, 2010. Cited on page 24.

[115] G. Hughes and L. Carr. Microsoft Smart Tags: Support, ignore or condemn them? In *Proceedings of the 13th ACM conference on Hypertext and hypermedia*, pages 80–81. ACM, 2002. Cited on page 23.

[116] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *SIGIR'93: Pro-*

ceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 329–338. ACM, 1993. Cited on pages 30 and 31.

[117] Institute of Medicine. *To Err is Human: Building a Safer Health System*. National Academy Press, 2000. Cited on page 147.

[118] Institute of Medicine. *Patient Safety: Achieving a New Standard for Care*. National Academy Press, 2004. Cited on page 147.

[119] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. Cited on page 183.

[120] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31: 264–323, 1999. Cited on pages 18 and 183.

[121] N. Jardine and C. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240, 1971. Cited on pages 6, 18, 19, 84, and 100.

[122] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, 2002. Cited on page 28.

[123] L. Jarvis and B. Stanberry. Teleradiology: threat or opportunity? *Clinical Radiology*, 60:840–845, 2009. Cited on page 147.

[124] F. Jelinek. *Statistical methods for speech recognition*. MIT Press, 1997. Cited on page 15.

[125] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980. Cited on page 16.

[126] V. Jijkoun, M. A. Khalid, M. Marx, and M. de Rijke. Named entity normalization in user generated content. In *AND'08: Proceedings of the second workshop on analytics for noisy unstructured text data*, pages 23–30. ACM, 2008. Cited on pages 23 and 149.

[127] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. Cited on page 138.

[128] S. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. Cited on pages 87 and 183.

[129] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL 2002 workshop on Natural language processing in the biomedical domain*, volume 3, pages 1–8, 2002. Cited on pages 151 and 155.

[130] M. Kc, R. Chau, M. Hagenbuchner, A. C. Tsoi, and V. Lee. A machine learning approach to link prediction for interlinked documents. In *INEX'09*, pages 342–354. Springer-Verlag, 2010. Cited on page 24.

[131] E. M. Keen. Presenting results of experimental retrieval comparisons. *Information Processing & Management*, 28:491–502, 1992. Cited on page 30.

[132] E. M. Keen and J. A. Digger. Report of an information Science Index Languages Test. *Aberystwyth, Department of Information Retrieval Studies, College of Librarianship Wales*, 1972. Cited on page 12.

[133] B. King. Step-Wise Clustering Procedures. *Journal of the American Statistical Association*, 62 (317):86–101, 1967. Cited on page 183.

[134] M. Krauthammer and G. Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526, 2004. Cited on page 151.

[135] M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman. Using blast for identifying gene and protein names in journal articles. *Gene*, 259(1-2):245–252, 2000. Cited on page 151.

[136] G. P. Krestin. Commoditization in radiology: threat or opportunity? *Radiology*, 256:338–342, 2010. Cited on page 147.

[137] O. Kurland. *Inter-document similarities, language models, and ad hoc information retrieval*. PhD thesis, Cornell University, 2006. Cited on pages 6, 18, 19, and 100.

[138] O. Kurland. The opposite of smoothing: a language model approach to ranking query-specific document clusters. In *SIGIR'08: Proceedings of the 31th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–178, 2008. Cited

on pages 18 and 20.

[139] O. Kurland. Re-ranking search results using language models of query-specific clusters. *Information Retrieval*, 12(4):437–460, 2009. Cited on pages 18 and 20.

[140] O. Kurland and C. Domshlak. A rank-aggregation approach to searching for optimal query-specific clusters. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 547–554. ACM, 2008. Cited on pages 6, 18, 19, 20, and 83.

[141] O. Kurland and L. Lee. 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. Cited on pages 16 and 19.

[142] O. Kurland and L. Lee. Respect my authority!: Hits without hyperlinks, utilizing cluster-based language models. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90. ACM, 2006. Cited on page 20.

[143] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119. ACM, 2001. Cited on page 16.

[144] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01: Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001. Cited on page 155.

[145] F. W. Lancaster. MEDLARS: report on the evaluation of its operating efficiency. *American Documentation*, 20(2):119–142, 1969. Cited on page 12.

[146] M. Larson, M. Tsagkias, J. He, and M. de Rijke. Investigating the global semantic impact of speech recognition error on spoken content collections. In *ECIR'09: Proceedings of the 31th European Conference on IR Research*, pages 755–760, 2009. Cited on page 9.

[147] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127. ACM Press, 2001. Cited on page 52.

[148] K.-J. Lee, Y.-S. Hwang, and H.-C. Rim. Two-phase biomedical ne recognition based on svms. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, volume 13, pages 33–40, 2003. Cited on pages 151 and 155.

[149] K. S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR'08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 235–242. ACM, 2008. Cited on page 20.

[150] W.-L. Lee and A. Lommatzsch. Feed distillation using adaboost and topic maps. In *TREC '07 Working Notes*. NIST, 2007. Cited on page 21.

[151] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC'86: Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986. Cited on page 153.

[152] G. Lidstone. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192, 1920. Cited on page 159.

[153] R. Ling. An exact probability distribution on the connectivity of random graphs. *Journal of Mathematical Psychology*, 12(1):90–98, February 1975. Cited on page 42.

[154] R. F. Ling and G. G. Killough. Probability tables for cluster analysis based on a theory of random graphs. *Journal of the American Statistical Association*, 71(354), 1976. Cited on page 42.

[155] X. Liu and W. B. Croft. 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. Cited on pages 6, 18, and 19.

[156] X. Liu and W. B. Croft. Experiments on retrieval of optimal clusters. Technical report, Center for

Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006. Cited on page 18.

[157] X. Liu and W. B. Croft. Representing clusters for retrieval. In *SIGIR'06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 671–672, 2006. Cited on page 18.

[158] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *ECIR'08: Proceedings of the 30th European Conference in Information Retrieval*, pages 454–462, 2008. Cited on page 18.

[159] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2:159–165, 1958. Cited on page 13.

[160] C. Macdonald and I. Ounis. Key blog distillation: ranking aggregates. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1043–1052. ACM, 2008. Cited on page 21.

[161] C. Macdonald and I. Ounis. The TREC Blogs06 collection: Creating and analyzing a blog test collection. Technical Report TR-2006-224, Department of Computer Science, University of Glasgow, 2006. Cited on page 57.

[162] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 Blog Track. In *TREC '07 Working Notes*, pages 31–43. NIST, 2007. Cited on pages 20, 21, 45, and 57.

[163] D. J. C. Mackay and L. Peto. A hierarchical dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1994. Cited on page 58.

[164] I. Mani. *Automatic summarization*. John Benjamins Publishing Co., 2001. Cited on page 3.

[165] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of mathematical statistics*, 18(1):50–60, 1947. Cited on page 87.

[166] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. Cited on pages 141 and 159.

[167] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Cited on pages 3, 11, 14, 18, 26, 27, 28, and 92.

[168] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7:216–244, 1960. Cited on pages 13 and 14.

[169] C. D. Maynard. Radiologists: Physicians or expert image interpreters? *Radiology*, 248:333–336, 2008. Cited on pages 147 and 148.

[170] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, S6, 2005. Cited on pages 151 and 155.

[171] E. Meij. *Combining concepts and language models for information access*. PhD thesis, University of Amsterdam, 2010. Cited on page 181.

[172] E. Meij and M. de Rijke. Thesaurus-based feedback to support mixed search and browsing environments. In *ECDL'07: Proceedings of the 11th European Conference on Digital Libraries*, 2007. Cited on page 12.

[173] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Learning semantic query suggestions. In *ISWC'09: Proceedings of the 8th International Semantic Web Conference*, pages 424–440, 2009. Cited on pages 12, 23, and 149.

[174] D. Metzler and W. B. Croft. 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*, pages 472–479. ACM, 2005. Cited on pages 87 and 141.

[175] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM'07: Proceedings of the sixteenth ACM conference on information and knowledge management*, pages 233–242. ACM, 2007. Cited on pages 7, 23, 24, 149, 152, 153, 154, and 162.

[176] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *SIGIR'99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221. ACM, 1999. Cited on page 15.

[177] G. A. Miller and E. B. Newman. Tests of a statistical explanation of the rank-frequency relation

for words in written english. *American Journal of Psychology*, 71:209–218, 1958. Cited on page 169.

[178] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM '08: Proceedings of the ACM 17th Conference on Information and Knowledge Management*, 2008. Cited on pages 7, 23, 24, 135, 139, 141, 149, 152, 153, and 154.

[179] G. Mishne. *Applied Text Analytics for Blogs*. PhD thesis, University of Amsterdam, 2007. Cited on page 21.

[180] G. Mishne and M. de Rijke. A study of blog search. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *ECIR'06: Proceedings of the 28th European Conference on Information Retrieval*, volume 3936, pages 289–301, 2006. Cited on page 46.

[181] S. Mizzaro. How many relevances in information retrieval? *Interacting With Computers*, 10: 305–322, 1998. Cited on pages 2 and 20.

[182] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Comput. Linguist.*, 17(1):21–48, 1991. Cited on page 48.

[183] W. Mossberg. New Windows XP feature can re-edit others' sites. *The Wall Street Journal*, 2001. Cited on page 23.

[184] S. S. Naik, A. Hanbidge, and S. R. Wilson. Radiology reports: examining radiologist and clinician preferences regarding style and content. *Am J Roentgenol*, 176:591–598, 2001. Cited on page 148.

[185] N. Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs), 2007. Cited on page 163.

[186] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *OSIR'6: Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval*, 2006. Cited on page 73.

[187] I. Ounis, C. Macdonald, M. de Rijke, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *TREC '06 Working Notes*. NIST, 2007. Cited on page 21.

[188] C. D. Paice. Soft evaluation of boolean search queries in information retrieval systems. *Inf. Technol. Res. Dev. Appl.*, 3:33–41, 1984. Cited on page 13.

[189] Y. Pilpel, P. Sudarsanam, and G. M. Church. Identifying regulatory networks by combinatiorial analysis of promoter elements. *Nat. Genet.*, 29:153–159, 2001. Cited on page 36.

[190] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998. Cited on pages 15 and 158.

[191] S. Preece. Clustering as an output option. In *Proceedings of the American Society for Information Science*, pages 189–190, 1973. Cited on page 19.

[192] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pages 784–791, 2008. Cited on pages 22 and 99.

[193] B. I. Reiner. The challenges, opportunities, and imperatives of structured reporting in medical imaging. *Journal of Digital Imaging*, 22(6):562–568, 2009. Cited on page 148.

[194] B. I. Reiner and E. L. Siegel. Decommoditizing radiology. *J Am Coll Radiol*, 6:167–170, 2009. Cited on page 147.

[195] B. I. Reiner, N. Knight, and E. L. Siegel. Radiology reporting: past, present, and future: the radiologist's perspective. *J Am Coll Radiol*, 4:313–319, 2007. Cited on page 148.

[196] S. Robertson. On the history of evaluation in IR. *Journal of Information Science*, 34(4):439–456, 2008. Cited on page 25.

[197] S. Robertson and J. Callan. Routing and filtering. In *TREC '05*, pages 99–122. NIST, 2005. Cited on page 21.

[198] S. E. Robertson. The probability ranking principle in IR. In *Readings in information retrieval*, pages 281–286. Morgan Kaufmann Publishers Inc., 1997. Cited on pages 14 and 99.

[199] S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*,

27(3):129–146, 1976. Cited on pages 13 and 15.

[200] S. E. Robertson and K. Spärck Jones. Simple proven approaches to text retrieval. Technical report, Computer Laboratory, Cambridge University, 1997. Cited on page 14.

[201] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc., 1994. Cited on page 15.

[202] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC'94: Proceedings of the 3rd Text REtrieval Conference*, 1994. Cited on page 73.

[203] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at trec-4. In *TREC'95: Proceedings of the 4th Text REtrieval Conference*, 1995. Cited on page 73.

[204] S. E. Robertson, E. Kanoulas, and E. Yilmaz. Extending average precision to graded relevance judgments. In *SIGIR'10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610. ACM, 2010. Cited on page 28.

[205] T. Roelleke and J. Wang. Tf-idf uncovered: a study of theories and probabilities. In *SIGIR'08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2008. Cited on page 15.

[206] P. Ruch. Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, 22(6):658–664, 2006. Cited on page 151.

[207] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968. Cited on pages 1 and 19.

[208] G. Salton. Cluster search strategies and the optimization of retrieval effectiveness. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 223–242. Prentice-Hall, 1971. Cited on page 18.

[209] G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971. Cited on pages 12, 13, and 19.

[210] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, 1973. Cited on page 13.

[211] G. Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. Technical report, Cornell University, 1974. Cited on page 13.

[212] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Commun. ACM*, 26: 1022–1036, 1983. Cited on page 13.

[213] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4:247–375, 2010. Cited on page 25.

[214] M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR'05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169. ACM, 2005. Cited on page 30.

[215] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *WWW'10: Proceedings of the 19th international conference on World wide web*, pages 881–890, 2010. Cited on pages 23 and 99.

[216] T. Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. part II: Nature and manifestations of relevance. *Journal of American Society for Information Science and Technology*, 58(13):1915–1933, 2007. Cited on page 1.

[217] T. Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. *Journal of American Society for Information Science and Technology*, 26 (6):321–343, 1975. Cited on page 1.

[218] L. Schamber, M. B. Eisenberg, and M. S. Nilan. A re-examination of relevance: toward a dynamic, situational definition. *Information Processing & Management*, 26(6):755–776, 1990. Cited on pages 1 and 2.

[219] R. Schenkel, F. Suchanek, and G. Kasneci. YAWN: A semantically annotated Wikipedia XML

corpus. In *BTW2007: Datenbanksysteme in Business, Technologie und Web*, 2007. Cited on pages 136 and 162.

[220] M. J. Schuemie, B. Mons, M. Weeber, and J. A. Kors. Evaluation of techniques for increasing recall in a dictionary approach to gene and protein name identification. *Journal of Biomedical Informatics*, 40(3):316–324, 2007. Cited on page 151.

[221] K. Seki, Y. Kino, and S. Sato. TREC 2007 Blog Track Experiments at Kobe University. In *TREC '07 Working Notes*. NIST, 2007. Cited on page 21.

[222] J. Seo and W. B. Croft. Blog site search using resource selection. In *CIKM'08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1053–1062. ACM, 2008. Cited on page 22.

[223] J. Seo and W. B. Croft. UMass at TREC 2007 Blog Distillation Task. In *TREC'07*, 2007. Cited on pages 21 and 22.

[224] B. Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *JNLPBA'04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. ACL, 2004. Cited on pages 151, 155, and 156.

[225] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 3(4):591–611, 1965. Cited on page 87.

[226] M. Shuemie, R. Jelier, and J. Kors. Peregrine: Lightweight gene name normalization by dictionary lookup. In *Second BioCreative Workshop*, pages 131–133, 2007. Cited on page 151.

[227] C. Silverstein and J. O. Pedersen. Almost-constant-time clustering of arbitrary corpus subsets4. In *SIGIR'97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 60–66. ACM, 1997. Cited on page 19.

[228] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR'96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM, 1996. Cited on page 15.

[229] P. K. C. Singitham, M. S. Mahabhashyam, and P. Raghavan. Efficiency-quality tradeoffs for vector score aggregation. In *VLDB'04: Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, pages 624–635. VLDB Endowment, 2004. Cited on page 18.

[230] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM'07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM, 2007. Cited on page 30.

[231] P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy: The principles and practice of numerical classification.* W.H. Freeman, 1973. Cited on pages 88 and 183.

[232] I. Soboroff. Does wt10g look like the web? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002. Cited on page 74.

[233] E. Soysal, I. Ciceklib, and N. Baykalc. Design and evaluation of an ontology based information extraction system for radiological reports. *Computers in Biology and Medicine*, 40(11-12):900–911, 2010. Cited on page 152.

[234] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. Cited on pages 13, 15, and 73.

[235] K. Spärck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36:779–808, 2000. Cited on page 14.

[236] N. Stokes, E. Newman, J. Carthy, and A. F. Smeaton. Broadcast news gisting using lexical cohesion analysis. In *ECIR'04: Proceedings of the 26th European Conference on Information Retrieval*, pages 209–222. Springer-Verlag, 2004. Cited on pages 48 and 50.

[237] D. R. Swanson. Some unexplained aspects of the cranfield tests of indexing performance factors. *Library Quarterly*, 41(3):223–228, 1971. Cited on page 1.

[238] K. Takeuchi and N. Collier. Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine*, 33(2):125–137, 2005. Cited on pages 151 and 155.

[239] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *HLT-NAACL'06: Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006. Cited on page 20.

[240] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of Pacific Symposium on Biocomputations*, pages 541–552, 2000. Cited on page 151.

[241] A. Tombros, R. Villa, and C. J. van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Inf. Process. Manage.*, 38(4):559–582, 2002. Cited on pages 18, 19, and 100.

[242] D. Trieschnigg, P. Pezik, V. Lee, F. de Jong, W. Kraaij, and D. Rebholz-Schuhmann. Mesh up: effective mesh text classification for improved document retrieval. *Bioinformatics*, 25(11): 1412–1418, 2009. Cited on page 151.

[243] Y. Tsuruoka and J. Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13*, pages 41–48. ACL, 2003. Cited on page 151.

[244] Y. Tsuruoka and J. Tsujii. Improving the performance of dictionary-based approaches in protein name recognition. *Journal of Biomedical Informatics*, 37(6):461–470, 2004. Cited on page 151.

[245] C. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. Cited on pages 6, 12, 18, 27, 82, and 84.

[246] C. J. van Rijsbergen and K. Spärck Jones. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *document information*, 29(3):251–257, 1973. Cited on page 18.

[247] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *CHI'04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582. ACM, 2004. Cited on page 149.

[248] E. Voorhees and D. Harman. Overview of the sixth text retreival conference. In *TREC'97: Proceedings of the 6th Text REtrieval*, 1997. Cited on page 73.

[249] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR'93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180. ACM, 1993. Cited on page 12.

[250] E. M. Voorhees. The TREC robust retrieval track. *SIGIR Forum*, 39:11–20, 2005. Cited on page 69.

[251] E. M. Voorhees. The cluster hypothesis revisited. In *SIGIR'85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 188–196. ACM, 1985. Cited on pages 18 and 19.

[252] E. M. Voorhees and D. Harman, editors. *TREC: Experiments and evaluation in information retrieval*. MIT Press, Cambridge, MA, 2005. Cited on pages 20 and 25.

[253] J. H. Ward Jr. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. Cited on page 183.

[254] J. Weeds, D. Weir, and D. Mccarthy. Characterising measures of lexical distributional similarity. In *Proceedings of CoLing 2004*, 2004. Cited on page 180.

[255] W. Weerkamp and M. de Rijke. Credibility improves topical blog post retrieval. In *HLT-NAACL'08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 923–931. ACL, 2008. Cited on page 21.

[256] W. Weerkamp, K. Balog, and M. de Rijke. Finding key bloggers, one post at a time. In *ECAI'08: Proceeding of the 18th European Conference on Artificial Intelligence*, 2008. Cited on pages 22, 46, and 52.

[257] W. Weerkamp, K. Balog, and M. de Rijke. Blog feed search with a post index. *Information*

Retrieval Journal, To appear. Cited on page 22.

[258] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR'06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006. Cited on page 19.

[259] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. Cited on pages 30 and 87.

[260] P. Willett. Recent trends in hierarchic document clustering: a critical review. *Inf. Process. Manage.*, 24(5):577–597, 1988. Cited on page 18.

[261] P. Willett. Query specific automatic document classification. *International Forum on Information and Documentation*, 10:28–32, 1985. Cited on page 19.

[262] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *SIGIR'99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 254–261. ACM, 1999. Cited on page 16.

[263] L. Yang, D.-H. Ji, G. Zhou, N. Yu, and G. Xiao. Document re-ranking using cluster validation and label propagation. In *CIKM'06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 690–697, 2006. Cited on pages 6 and 18.

[264] Z. Yang, H. Lin, and Y. Li. Exploiting the performance of dictionary-based bio-entity name recognition in biomedical literature. *Computational Biology and Chemistry*, 32(4):298–291, 2008. Cited on page 151.

[265] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty. In *SIGIR'05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 512–519, 2005. Cited on pages 69 and 70.

[266] Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. In W. W. Cohen, A. Mccallum, S. T. Roweis, W. W. Cohen, A. Mccallum, and S. T. Roweis, editors, *ICML'08: Proceedings of the 25th international conference on Machine learning*, volume 307, pages 1224–1231. ACM, 2008. Cited on page 22.

[267] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54. ACM, 1998. Cited on page 19.

[268] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004. Cited on pages 16 and 57.

[269] C. Zhai and J. Lafferty. A risk minimization framework for information retrieval. *Information Processing & Management*, 42(1):31–55, 2006. Cited on page 22.

[270] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. ACM, 2003. Cited on pages 22 and 99.

[271] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM, 2002. Cited on page 37.

[272] Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *ECIR'08: Proceedings of 30th European Conference on IR Research on advances in Information Retrieval*, pages 52–64, 2008. Cited on page 69.

[273] G. Zhou, J. Zhang, J. Su, D. Shen, and C. Tan. Recognizing names in biomedical texts: A machine learning approach. *Bioinformatics*, 20:1178–1190, 2004. Cited on pages 151 and 155.

[274] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 1998. Cited on page 25.

Samenvatting

Het gebruik van thematische informatie wordt al lang bestudeerd in het vakgebied van Information Retrieval. Het groeperen van zoekresultaten in thematische categorieën leidt bijvoorbeeld tot een meer effectieve presentatie van informatie aan de gebruiker, terwijl het groeperen van documenten in een collectie voor meer efficiënte toegang tot informatie kan zorgen. We definiëren thema als het belangrijkste onderwerp in een (verzameling van) document(en). Terwijl thema's informatie verschaffen over de onderwerpen die in een document aan bod komen, geeft de structuur van thema's ons informatie over de mate waarin een verzameling documenten is gericht op bepaalde thema's, de diversiteit van thema's in documenten, en de semantische relaties tussen thema's.

Het werk in dit proefschrift richt zich op het modelleren van de structuur van thema's. In het bijzonder bekijken wij een aantal IR taken waarin de notie van relevantie verder gaat dan "aboutness" en waarin de structuur van thema's een belangrijke rol speelt in het vervullen van de informatiebehoefte van gebruikers. De volgende onderzoeksonderwerpen komen aan bod: (1) Thema-coherentie; hier ontwikkelen we een coherentiescore die effectief de thematische samenhang van een verzameling documenten beschrijft. Deze score wordt toegepast op twee IR taken, namelijk, *blog feed retrieval* en *query performance prediction*. (2) Diversiteit en de clusterhypothese; hier onderzoeken we de relatie tussen diversiteit, relevantie en de cluster hypothese. We werpen opnieuw een blik op de clusterhypothese maar nu in relatie tot ambigue vragen of vragen met meerdere aspecten en onderzoeken de effectiviteit van zoekvraagspecifieke clustering voor het diversifiëren van zoekresultaten. (3) Het verbinden van thema's in verschillende representaties. Thema's kunnen op verschillende manieren worden gerepresenteerd, bijvoorbeeld, door middel van clusters, definities in een thesaurus en statistieken over term frequenties. We bestuderen het probleem van het verbinden van thema's die worden gerepresenteerd in verschillende vormen in de context van automatische linkgeneratie. We identificeren significante termen in een brontekst en linken deze termen aan corresponderende items in een kennisbank gevuld met achtergrondinformatie over deze termen.

Abstract

The use of topical information has long been studied in the context of information retrieval. For example, grouping search results into topical categories enables more effective information presentation to users, while grouping documents in a collection can lead to efficient information access. We define a *topic* as the main theme or subject contained in a (set of) document(s). While topics provide information about the subjects contained in a document, the structure of topics provides information such as the degree to which a set of documents is focused on certain topic (or set of topics), topical diversity among documents, and semantic relatedness of topics.

The work of this thesis focuses on modeling the *structure* of topics present in a (set of) document(s), with the goal of effectively using it in information retrieval. In particular, we consider a number of IR tasks where the notion of relevance is beyond "aboutness" and topic structure plays an important role in satisfying users' information need. The following research themes are addressed: (1) Topic coherence; here we develop a coherence score that effectively captures topical coherence of a set of documents. The proposed score is applied to two IR tasks, namely, blog feed retrieval and query performance prediction. (2) Diversity and the cluster hypothesis, where we investigate the relation between diversity, relevance and the cluster hypothesis. We re-visit the cluster hypothesis with respect to ambiguous or multi-faceted queries and investigate the effectiveness of query-specific clustering in result diversification. (3) Relating topics present in different representations. Topics can be represented in different ways, e.g., using clusters, using definitions from a thesaurus, using statistics of term frequencies, etc. We study the problem of relating topics represented in different forms within the context of automatic link generation. We identify a set of significant terms from a source text, link those terms to their corresponding entries in a knowledge base in such a way that the source text is annotated with background information available in the knowledge base.