Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)
http://dare.uva.nl/document/216751

File ID	216751
Filename	Chapter 2: Background

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	Dissertation
Title	Exploring topic structure: coherence, diversity and relatedness
Author	J. He
Faculty	Faculty of Science
Year	2011
Pages	xi, 201
ISBN	978-94-90371-81-4

FULL BIBLIOGRAPHIC DETAILS:
http://dare.uva.nl/record/377895

Copyright

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other then for strictly personal, individual use.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)

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.