Exploiting Social Semantics for Multilingual Information Retrieval

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)
von der Fakultät für
Wirtschaftswissenschaften
am Karlsruher Institut für Technologie

vorgelegte Dissertation

von

Dipl.-Inf. Philipp Sorg

Tag der mündlichen Prüfung:	22. Juli 2011
Referent:	Prof. Dr. Rudi Studer
Korreferent:	Prof. Dr. Philipp Cimiano
Prüfer:	Prof. Dr. Andreas Oberweis
Vorsitzender der Prüfungskommission:	Prof. Dr. Christof Weinhardt

Abstract

Information Retrieval (IR) deals with delivering relevant information items given the specific information needs of users. As retrieval problems are defined in various environments such as the World Wide Web, corporate knowledge bases or even personal desktops, IR is an every day problem that concerns almost everybody in our society. In this thesis, we present research results on the problem of Multilingual IR (MLIR), which defines retrieval scenarios that cross language borders. MLIR is a real-world problem which we motivate using different application scenarios, for example search systems having users with reading skills in several languages or expert retrieval.

As the main topic of this thesis, we consider how user-generated content that is assembled by different popular Web portals can be exploited for MLIR. These portals, prominent examples are Wikipedia or Yahoo! Answers, are built from the contributions of millions of users. We define the knowledge that can be derived from such portals as *Social Semantics*. Further, we identify important features of Social Semantics, namely the support of multiple languages, the broad coverage of topics and the ability to adapt to new topics. Based on these features, we argue that Social Semantics can be exploited as background knowledge to support multilingual retrieval systems.

Our main contribution is the integration of Social Semantics into multilingual retrieval models. Thereby, we present Cross-lingual Explicit Semantic Analysis, a semantic document representation that is based on interlingual concepts exploited from Wikipedia. Further, we propose a mixture language model that integrates different sources of evidence, including the knowledge encoded in the category structure of Yahoo! Answers.

For evaluation, we measure the benefit of the proposed retrieval models that exploit Social Semantics. In our experiments, we apply these models to different established datasets, which allows for the comparison to standard IR baselines and to related approaches that are based on different kinds of background knowledge. As standardized settings were not available for all the scenarios we considered, in particular for multilingual Expert Retrieval, we further organized an international retrieval challenge that allowed the evaluation of our proposed retrieval models which were not covered by existing challenges.

Contents

Chapter I

Introduction

The field of Information Retrieval (IR) is concerned with satisfying information needs of users. The IR approach therefore is to find and to present information items, for example documents, that contain the relevant information. IR covers various application scenarios — related to our work life as well as to our leisure activities. It is no exaggeration to say that IR is an every day problem that concerns almost everybody in our society. The most prominent example is certainly searching the World Wide Web. The sheer mass of websites requires efficient approaches to retrieve relevant subsets for specific information needs. However, a constantly increasing number of information items are also gathered in corporate knowledge bases or even on our personal computers. This requires to adapt the retrieval techniques applied to Web search to these new scenarios.

Many of these information items — for example websites, posts to social networks or personal emails — are written in different languages. In fact, only one fourth of Internet users are native English speakers.[1] The nature of the Internet does not know any language boundaries. People from different nations and languages are for example connected in social networks. This clearly motivates the development and improvement of multilingual methods for IR, which also cross the language barriers when seeking for information. Users may often be interested in relevant information in different languages, which are retrieved in a single search process when using multilingual technologies. This also allows users to express the information need in their mother tongue while retrieving results in other languages.

The bottleneck for the development of multilingual approaches to IR are language resources that mediate between the languages. Examples for such resources that are often used in current multilingual retrieval systems are bilingual dictionaries or interlingual wordnets such as EuroWordNet[2]. These traditional resources are usually hand crafted and cover only a limited set of topics. As these are closed systems,

[1] There are 27.3% of English Internet users according to `http://www.internetworldstats.com` (last accessed November 16, 2010)

[2] `http://www.illc.uva.nl/EuroWordNet/` (last accessed April 8, 2011)

they depend on revisions for updates — which are usually expensive and therefore infrequent. In this thesis, we propose to explore new types of multilingual resources. Evolving from the Web 2.0, we define *Social Semantics* as the aggregated knowledge exploited from the contributions of millions of users. Using Social Semantics for IR has several advantages compared to using traditional multilingual language resources. First of all, many languages and many domains are covered as people from all over the world contribute to Social Web sites about almost any topic. These resources are thereby constantly growing. This has also the consequence that they are up-to-date as they almost instantly adapt to new topics.

The questions remains how resources of Social Semantics can be exploited for multilingual retrieval — which is the central research question behind this thesis. We show that these collaboratively created datasets have many features that can be explored in respect to their application to IR.

In this section, we first describe and motivate multilingual retrieval scenarios. Then, we will present the definition of semantics that is used throughout this thesis. We will also define IR — in particular cross-lingual and multilingual retrieval. Following these definitions, we will present the main research questions that are considered in this thesis. This includes a summary of our contributions in respect to these research questions. Finally, we will give an overview of all chapters that guides through the content of this thesis.

I.1 Multilingual Retrieval Scenario

The question might be raised whether there is a real need for multilingual retrieval. We will motivate the investigation in this multilingual problem by the following two scenarios.

Internet usage statistics as presented in Figure I.1 show that only one fourth of the Internet users are native English speakers. English speakers still constitute the biggest user group, but this is likely to change as Internet penetration — which is almost saturated in most English speaking countries — will grow for example in Chinese or Spanish speaking areas. It can be assumed that many of these users are able to read more than one language. In the European Union for example, more than half of the citizens assert that they can speak at least one other language than their mother tongue [TNS Opinion & Social, 2005]. The results of the according survey are presented in Figure I.2.

As a consequence of the ability to understand more than one language, these users are also interested in Web content of different languages which motivates multilingual retrieval scenarios. Multilingual users will probably be most confident in formulating their information need using their mother tongue. However, they will be interested in relevant information in any language they are able to understand. This gives real benefit in cases when relevant resources are scare in the original query language.

The second scenario corroborating the need for multilingual retrieval is Entity

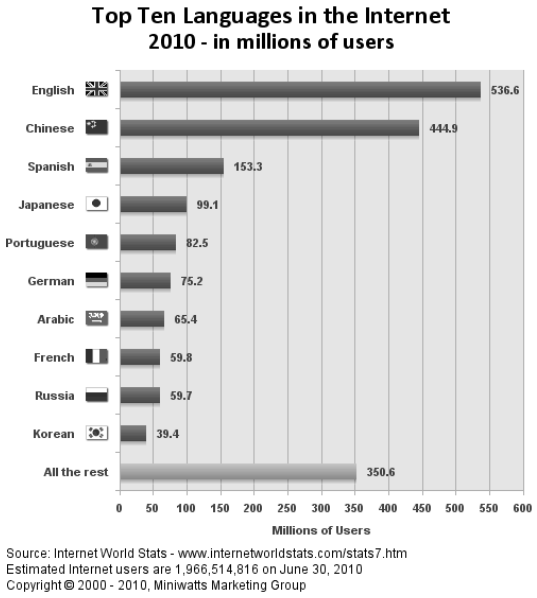


Figure I.1: Statistics of the number of Internet users by language.

Search. In contrast to document retrieval, the items returned by the retrieval system are entities which are not as dependent on a specific language as text documents are. Examples for entity classes are locations or people. In the search process, evidence of different languages can be used to identify relevant entities. In this thesis, we will present research results on Expert Retrieval as an application of multilingual Entity Search.

An argument against multilingual retrieval scenarios is the usage of English as *lingua franca* of the Internet. Assuming that all relevant information is available in English and can be read by all Internet users makes multilingual approaches obsolete. However, the statistics presented in Figure I.1 motivate a different view on the language landscape of the Internet. Information is published in various languages and many users will not be able to access the English part of the Internet. It is therefore important to consider other languages as English as well and to develop multilingual retrieval methods that allow to cross the boundaries of languages.

Many multilingual retrieval approaches are based on Machine Translation (MT) systems. These systems allow the automatic translation of text to another language. In the context of retrieval, MT systems have the potential to reduce the problem of multilingual retrieval to the monolingual case. However, these systems still have high error rates — which are for example rooted in the ambiguity of terms or in the complexity of grammar. While all IR systems have to deal with the ambiguity of terms, the problem is that errors introduced in the translation step are potentially

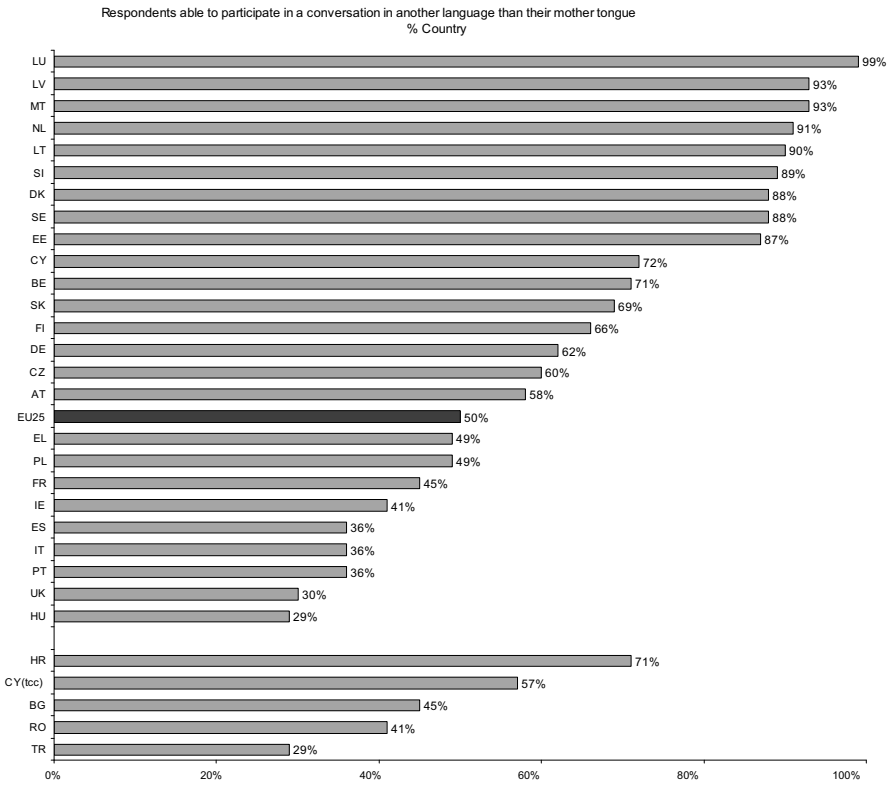


Figure I.2: Statistics of the share of citizens in the European Union that are able to understand another language aside from their mother tongue. Source: *Europeans and Languages*, Eurobarometer Special Survey 237, Wave 63.4, TNS Opinion & Social, May-June 2005.

amplified in retrieval systems that are based on MT. Further, only few popular languages are supported by current MT systems. These problems will most probably not be solved in the near future. This motivates the development of multilingual retrieval methods that do not depend on MT or at least are able to compensate errors introduced by the translation systems.

I.2 Definition of Semantics

In this thesis, we present new approaches to exploit Social Semantics for MLIR. Different definitions of the term *semantics* are established in literature — depending on the research field and specific context. In the following, we will give an overview of these different definitions and precisely define the term *semantics* in the context of this thesis.

I.2.1 Historical Overview

The term *semantics* was introduced by M. Bréal at the beginning of the 20th century to describe the subfield of linguistics that analyzes and describes the *meaning* of linguistic expressions [Bußmann, 1983]. Different focus is thereby set on the meaning of single lexicon tokens, relations between those tokens (for example synonymy), the meaning of sentences as the sum of meaning of its single lexemes as well as their grammatical relation and the relation of linguistic expressions and their meaning to the physical world.

As part of semantics, the research field of *formal semantics* adapts analytical techniques from logic to natural language [Abbott, 1999]. The goal thereby is to find formal and precise definitions of the meaning of linguistic expressions. Abbott [1999] gives an overview of the historical development of formal semantics. As one of the first researchers in this field, Bloomfield formally defined semantics by stimulus and response. The meaning of text depends on the current situation of the speaker and the response she gets from hearers. Therefore the knowledge about everything in the speakers' world is required to define the meaning of text. This definition of formal semantics was impractical and the research did not advance till the introduction of generative models of language by Chomsky. However, Chomsky himself was convinced that intuition about meaning could not be used to improve the analysis of linguistic forms. He insisted upon the radical autonomy of syntax, *i.e.* syntactic rules are the initial stage of language that is not influenced by meaning [Joseph, 2002]. As a reaction to this hypothesis, Montague disagreed and claimed that there is no difference between natural languages and formal languages. In fact, semantic and syntactic representations mirror each other. This theory was the advent of recent approaches to formal semantics based on truth conditions. These truth conditions, which are equivalent to models used in logic, are existential to define the semantics of linguistic expressions. Knowing the meaning implies knowing the truth conditions which therefore relate language to the outside world.

Apart from linguistics, the semantics of text or documents is also subject of investigation in computer science. The vision of the *Semantic Web* [Berners-Lee et al., 2001] includes the formal semantic description of data that makes it machine processable. The Semantic Web is build on ontologies [Studer et al., 1998] that formalize vocabulary by logics with clearly defined semantics. The meaning of the vocabulary modeled in ontologies can therefore be shared and can also be further processed using automatic reasoners. A common way to implement ontologies is to use description logics, which allows inference and the assignment of truth conditions using models analogous to Montague's theory on formal semantics.

The Semantic Web vision considers the Internet as a database providing structured knowledge with clearly defined semantics, which is also denoted as the *Web of Data*. However the biggest share of content is still provided in unstructured text with no explicit meaning, including most of the user generated content, which is denoted as the *Web of Documents*. Our definition of semantics is less precise but allows to describe the implicit knowledge found in these unstructured or semi-structured datasets. This will be clarified in the next section.

I.2.2 Social Semantics defined by Category Systems

Traditional semantic data sources have several drawbacks — especially in their application to open domain problems such as IR. As examples, ontologies often model a specific domain and therefore have a limited coverage. These ontologies potentially contain labels in different languages, but in most cases they are only defined in one specific language. Linguistic resources like thesauri are updated infrequently due to the high costs of new revisions. This motivates the exploitation of user-generated content as semantic data sources from the *Web 2.0*.

Definition I.1 (Web 2.0) *The* Web 2.0 *is defined by the interaction and socialization of users in the Internet. This includes all kind of user-generated content, for example collections of individual contributions or collaboratively created resources. As opposite to Web sites created by few editors, a large number of users — in many cases all Internet users — are allowed to contribute and provide new content.*

The datasets created in the scope of the Web 2.0 do not contain an axiomatic definition of vocabulary as found in the Semantic Web. In contrast, they contain aggregated knowledge of many users that we define as *Social Semantics*:

Definition I.2 (Social Semantics) *We define the implicit knowledge created by the contributions of many users as* Social Semantics*. The semantics of this aggregated data is defined through usage.* Concepts *defined by Social Semantics have no axiomatic definition. The collective usage of these concepts implies that there is an implicit agreement to understand them in the same way. This* bottom-up *semantics of concepts contrast to the logical definition of concepts in the Semantic Web.*

The usage of Social Semantics as background knowledge for MLIR is motivated by the properties of these datasets. Data sources that are exploited in respect to MLIR scenarios should support the following features — which are often present in Web 2.0 datasets:

Interlingual: In the application area of MLIR, data sources need to cover all supported languages. Ideally, they also provide connections between languages which help to cross the language boundaries in the retrieval process.

This is often given for Web 2.0 portals that support users from many different nations and languages.

Topic Coverage: General IR systems are not limited to special domains and potentially have to retrieve documents for any information need. IR systems based on background knowledge therefore require data sources covering a broad range of topics corresponding to the different information needs of the users.

Web 2.0 portals are usually also not limited in respect to domains and therefore have content about many topic fields. Additionally, this content is user generated and will most probably mirror the same popular domains that can also be found in the IR scenario.

Up-to-Date: IR systems need to adapt quickly to new topics as these are the topics users will most likely be searching for. Data sources used to enhance IR need to adapt to these topics as well. Only up-to-date and adaptive resources will improve search for emerging topics that were not covered before.

In the Web 2.0, content is constantly added that will most probably also cover the most recent topics. This means that datasets from Web 2.0 portals are updated very quickly and are therefore able to adapt to new topics.

We consider two prominent families of Web 2.0 sites in this thesis to define Social Semantics. These are *Wikis* and *Social Question/Answer Sites*. Both of them are instances of datasets that support the above mentioned properties and are therefore qualified as background knowledge in the MLIR scenario. In the following, we will define both Wikis and Social Question/Answer Sites (SQASs):

Definition I.3 (Wiki) Wikis *are collections of articles that allow users to collaboratively create, edit and comment these articles. As Wikis contain the aggregated knowledge of their users, they are often regarded as example of* Collective Intelligence. *The most prominent example of a Wiki covering many domains and languages is* Wikipedia.[3]

Definition I.4 (Social Question/Answer Site) Social Question/Answer Sites (SQASs) *allow users to post questions to the system. In contrast to IR, these*

[3] `http://www.wikipedia.org/` (last accessed April 8, 2011)

questions are not answered automatically based on some background collection but other users are able to provide answers. Most SQASs include social features like rating of questions or answers, which help to identify questions of high quality and appropriate answers. As example, the currently most popular SQAS is Yahoo! Answers.[4]

The question remains what kind of structures evolve from Web 2.0 sites that can be used as background knowledge for IR. In general, Web 2.0 sites do not define expressive ontologies. However, in many cases they feature a hierarchical organization of the user-generated content. We will define such Web 2.0 sites as *category systems*. An example for such category systems are Wikis or SQASs which allow the assignment of articles respectively questions to categories. Category systems provide the semantic structures we analyze in this thesis as instances of Social Semantics.

Definition I.5 (Category Systems) *We define* category systems *as hierarchical organizations of concepts.* Concepts *are defined by textual description in one or more languages.* Categories *correspond to organizational units that consist of clusters of related concepts and of sub-categories that allow a more detailed classification of concepts. Thereby, the classification of concepts to categories is no binary decision, concepts can be associated to several categories.*

Category systems do not define a formal hierarchy as the hierarchical relations between categories have different semantics. This includes for example the is-a *relation as used in taxonomies, but is not limited to it.*

We will show how category systems can be used as semantic resources to improve MLIR. They evolve from Web 2.0 sites and often support the properties of Social Semantics as defined above — they support many languages, cover a broad range of topics and are constantly updated.

I.3 Definition of Information Retrieval

Information Retrieval (IR) deals with the representation, storage, organization of and access to information items [Baeza-Yates and Ribeiro-Neto, 1999]. IR systems with suitable representation and organization of the information items allow users with an information need to have easy access to the information she is interested in.

Typically, the user manifests her information need in the form of a query — usually a bag of keywords — to convey her need to the IR system. The IR system will then retrieve items which it believes are relevant to the user's information need. Indeed, the notion of relevance is at the center of IR. It is defined on the semantic level and defines whether the content of a retrieved item satisfies the user's information need. The users's satisfaction with the IR system is linked to how quickly the user is able to find the relevant items. This implies that the retrieval of non-relevant items,

[4] `http://answers.yahoo.com/` (last accessed April 8, 2011)

particularly those ranked higher than the relevant items, represent a less than satisfactory retrieval outcome for the user. We will use the following definition for IR throughout this paper:

Definition I.6 (Information Retrieval) *Given a collection D containing information items d_i and a keyword query q representing an information need, IR is defined as the task of retrieving a ranked list of information items $d_1, d_2, \ldots$ sorted by their relevance in respect to the specified information need.*

In the monolingual case, the content of information items d_i and the keyword query q are thereby written in the same language.

Related to IR is the problem of *Question Answering*. The main distinction of these two problems is given by the different types of information needs and the presentation of results. In IR, the information need is often imprecise or vague. IR systems are mostly used as explorative systems that allow users to iteratively formulate their information need based on preliminary results. In contrast, Question Answering requires sharp and clearly expressed information needs. Further, the presentation of a ranked list of items that contain relevant information is different. In Question Answering systems, single facts or short explanations are expected that precisely answer the given information need. Using IR systems, users have to extract these facts and explanations from the list of relevant items themselves. However, many Question Answering approaches build on IR by mining the ranked list of information items for possible answers (see for example [Kwok et al., 2001]).

I.3.1 Multilingual IR

In Cross-lingual and Multilingual IR, the information need and the corresponding query of the user may be formulated in other languages than the one in which the documents are written in. Relevance is in principle a language independent concept, as it is defined on the semantic level of both documents and information need.

Cross-lingual IR (CLIR) is the task of retrieving documents relevant to a given query in some language (query language) from a collection of documents in some other language (collection language).

Definition I.7 (Cross-lingual IR) *Given a collection D containing documents in language l_D (collection language), CLIR is defined as retrieving a ranked list of relevant documents for a query in language l_q (query language), with $l_D \neq l_q$. D is a monolingual collection — all documents in D have the same language.*

In contrast to CLIR, Multilingual IR (MLIR) considers corpora containing documents written in different languages. It can be defined as follows:

Definition I.8 (Multilingual IR) *Given a collection D containing documents in languages $l_1, \ldots, l_n$ with $l_i \neq l_j$ for $1 \leq i, j \leq n, i \neq j$, MLIR is defined as*

the task of retrieving a ranked list of relevant documents for a query in language l_q. These relevant documents may thereby be distributed over all languages $l_1, \ldots, l_n$.

MLIR finds application in all those settings where a dataset consists of documents in different languages and users of the retrieval system have reading capabilities in some of the languages the documents are written in. In most cases, people have indeed basic reading and understanding skills in some other language than their mother tongue (the one they usually query the collection). Such settings can be found in particular in the Web but also in large corporate multinationals. Further, still if the users do not understand the language of a returned document, Machine Translation systems can be applied to produce a text in the native language of the user.

I.3.2 Entity Search

Entity Search (ES) is a subfield of IR that attracts more and more attention. For example at TREC — the most important platform for IR evaluation — a complete track is dedicated to ES [Balog et al., 2009b]. Instead of documents, a set of *entities* is searched. In analogy to standard IR, ES systems are expected to return a ranked list of relevant entities given an information need. Examples of such entities are people (often referred to as *Expert Retrieval*) or other named entities (cities, countries, ...). In the context of this thesis, we extend ES to the multilingual scenario:

Definition I.9 (Multilingual Entity Search) *Given a set of entities E, Multilingual Entity Search (MLES) is defined as the task of retrieving a ranked list of relevant entities for a query in language l_q. Entities are information items that do not depend on a specific textual representation and have the same meaning in the context of any language. Examples for such entities are people or locations. Therefore, the notion of relevance is independent of the query language or the language of the textual evidence that is used to rank each entity.*

In many cases, the problem of MLES can be reduced to the problem of MLIR by defining a multilingual *text profile* of each entity. However, this is not a trivial step as the weighting of terms or the integration of different text sources might be difficult. Another problem in ES is the result presentation. While the names of entities might not be human readable or even missing, other *descriptive properties* of entities need to be extracted and presented to the user of the search system.

I.4 Research Questions

The main research question we address in this thesis is how semantic background knowledge can be exploited for MLIR. This problem consists of two parts.

The first problem is the discovery or selection of suitable resources. Given the retrieval scenario, these resources need to cover a broad range of topics and should be able to adapt to dynamic changes. In addition, the information provided by these

resources should be useful in respect to the retrieval scenario. We assume that there is a positive correlation between quality and usefulness, as resources of low quality will probably not be very helpful when used as background knowledge. Examples of low quality resources are resources containing a large share of noise, which makes it hard to identify meaningful content. Therefore, a high quality of data is also required for suitable resources.

Our proposed approaches to solve this problem aim at using datasets from the Web 2.0. We argue that commonly used semantic data sources such as domain ontologies do not meet the properties of Social Semantics in the general case, which are however preferable in the MLIR scenario. Our contribution is therefore the exploitation of collaboratively created Web 2.0 resources, which are built by the contributions of many Internet users and define implicit knowledge — the Social Semantics as defined in Definition I.2.

The second problem is how the semantic resources can be integrated in retrieval models. The main goal is to improve retrieval performance by using the knowledge provided by these resources. However, the integration must be robust in respect to changes in the background knowledge. Further, improvements in retrieval performance need also to be consistent when applied to different information needs and document collections. While the definition of retrieval models exploiting the background knowledge is the central task to solve this problem, focus has also to be set on experimental design and evaluation of the new models.

In summary, the main research questions and our approaches to solve these questions are the following:

Selection of Resources: *What are suitable resources that can be used as semantic background knowledge to support MLIR systems, and that also meet the properties of Social Semantics?*

In more detail, we analyzed two multilingual Web 2.0 portals in respect to the properties of Social Semantics: Wikipedia and Yahoo! Answers.

In the case of Wikipedia, a large coverage of topic fields is given. However, it is not clear if all content is linked across languages. We analyzed Wikipedia databased in several languages and created statistics that indeed show that the level of linkage across languages is sufficient and also of high quality [Sorg and Cimiano, 2008a]. We also identified missing cross-language links in Wikipedia [Sorg and Cimiano, 2008b]. In Chapter VI, we present results that show that most of these missing links were added by the Wikipedia community since our experiments — supporting the hypothesis that Wikipedia is constantly improving and adapting.

As regarding to Yahoo! Answers, we analyzed the distribution of questions to categories and languages, showing the large coverage of topic fields and also the support for various languages [Sorg et al., 2010]. The growing rate of Yahoo! Answers is much higher compared to Wikipedia due to the lower barriers of publishing content. This growing rate implies that Yahoo! Answers

is constantly updating and therefore also adapting to new topics.

Application to MLIR: *How can the knowledge encoded in these resources be used to support multilingual retrieval?*

We considered different approaches to answer this question.

Using Wikipedia as background knowledge, we analyzed how *Explicit Semantic Analysis*, an approach to concept indexing, can be extended to multilingual scenarios. Therefore we defined Cross-lingual Explicit Semantic Analysis that exploits Wikipedia databases in different languages and can be applied to CLIR and MLIR [Sorg and Cimiano, 2008a]. Then, we analyzed different design choices of Cross-lingual Explicit Semantic Analysis that allow the optimization of the multilingual retrieval system and the exploitation of different features of Wikipedia such as category links [Sorg and Cimiano, 2009, 2011a]. In our evaluation, we also addressed the question how Cross-lingual Explicit Semantic Analysis as a Wikipedia based retrieval system compares to related approaches relying on implicit concepts. Our results show that Cross-lingual Explicit Semantic Analysis achieves comparable results without expensive training on the test dataset [Cimiano et al., 2009].

In a second retrieval scenario on Yahoo! Answers, we addressed the question of how the knowledge given by the question/answer history can be used to improve a multilingual retrieval system. Our contributions are two-fold. Firstly, we show that the structural information encoded in the category classification of questions can be integrated into standard retrieval models. We therefore defined a mixture language model that is able to integrate different sources of evidence. This resulted in a significant improvement compared to the performance of several baselines in the Expert Retrieval scenario used in our evaluation. Secondly, we show that parameters of such integrated retrieval systems can be optimized using Machine Learning techniques. We used a discriminative model to design a retrieval system that is based on a trained classifier and input generated by various language models. Our results show that the values of several evaluation measures are significantly improved using the discriminative model that is trained on existing relevance assessments.

Evaluation: *What is the impact in respect to IR performance of the proposed semantic retrieval models compared to standard approaches?*

The evaluation of the proposed models requires datasets that allow the definition of cross-lingual or multilingual scenarios. In the case of Cross-lingual Explicit Semantic Analysis, we participated at an international IR challenge that defined a multilingual search task, including a standardized dataset, topics, relevance assessments and evaluation measures. The results show that Cross-lingual Explicit Semantic Analysis is comparable with competing retrieval systems for certain topic languages [Sorg et al., 2009].

To evaluate the retrieval system exploiting Yahoo! Answers, we used an Expert Retrieval scenario on this dataset. As we were first to propose such a scenario, no evaluation frameworks were available for these experiments. We therefore addressed the problem of creating a well defined MLIR challenge on top of the Yahoo! Answers dataset. We organized the CriES workshop at CLEF, including the creation of the ground truth in respect to a set of carefully selected topics [Sorg et al., 2010].

I.5 Overview of the Thesis

In Chapter II, we present preliminaries of IR. We will define IR and the search pipeline that is used in most IR systems. Further, we will define the specific problems addressed by IR. This chapter gives important background that is needed to understand the new models we propose in this thesis.

In Chapter III, we give an overview of how semantics are used in IR. This includes different definitions of semantics. We distinguish between three approaches to apply semantics to IR — semantic document models, semantic relatedness and semantic retrieval models. The intention of this chapter is to present the most relevant related work in respect to these different approaches.

In Chapter IV, we define Cross-lingual Explicit Semantic Analysis (CL-ESA). This model addresses all three research questions as defined in Section I.4. We describe how concepts can be extracted from Wikipedia that are aligned across languages. This is applied to multilingual retrieval by the definition of an interlingual concept space which is then used for CL-ESA. In this context, we present different design choices and parameters of CL-ESA that allow to optimize the retrieval model. Finally, this chapter also contains the evaluation of CL-ESA on several CLIR and MLIR scenarios.

In Chapter V, we present another approach to exploit Social Semantics. We present multilingual retrieval models that are based on the theory of language models and allow to combine different sources of evidence. This includes classification systems as given by the categories in Social Question/Answer Sites. Again, the three research questions are addressed. As resource we selected a dataset from Yahoo! Answers. We defined a retrieval model that is able to exploit this data to improve the retrieval performance in an Expert Retrieval scenario. To evaluate these models, we designed an official retrieval challenge which also allows to compare our models to alternative IR systems.

In Chapter VI, we present a Data Mining approach to detect missing cross-language links in Wikipedia. The quality of this Data Mining approach is evaluated using a test corpus. We also present long term studies of the development of these missing links in Wikipedia. This aims at the research question of selecting appropriate resources. Our results show that it is possible to improve Wikipedia using automatic approaches and that Wikipedia is actually improving over time. This qualifies Wikipedia as a resource for Social Semantics.

In Chapter VII, we summarize the outcomes of this thesis and give an outlook to future applications and research.

Chapter II

Preliminaries of Information Retrieval [1]

In this chapter, we present preliminaries of IR that help to understand the semantic retrieval models we present in this thesis. We focus on multilingual aspects and will give further background on common approaches to deal with documents and queries in different languages.

We already defined IR in Chapter I (see Definition I.6). Given the user information need in form of a query, IR systems return information items ranked by their relevance. The overall search process is visualized in Figure II.1. This process consists of two parts. The *indexing part* processes the entire document collection to built index structures that allow for efficient retrieval. These are usually inverted indexes. Each document is thereby preprocessed and mapped to a vector representation. In this thesis, we will discuss different approaches to define such representation, either based on terms or concepts. The *search part* is based on the same preprocessing step that is also applied to the query. Using the vector representation of the query, the matching algorithm determines relevant documents which are then returned as ranked results.

In detail, this chapter is structured as follows:

Document Preprocessing: We describe the preprocessing that can be applied to documents containing text in various languages. This is not only limited to languages using scripts based on the Latin alphabet, but also includes the preprocessing of documents containing text of other scripts and character encodings. We will also introduce specific approaches to solve the problems of tokenization and normalization that are required to process text depending on the used script and language.

Monolingual IR: We introduce different document models, retrieval models and

[1] This chapter is based on [Sorg and Cimiano, 2011b].

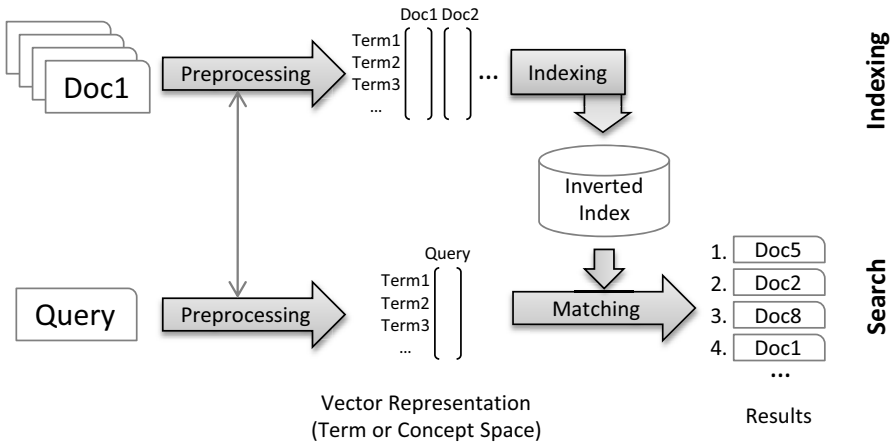


Figure II.1: The IR pipeline using vector space representation of documents and queries. The IR process consists of two processing pipelines: indexing of documents and searching of relevant documents given a query.

index structures that have been applied to IR. We will also present extensions to basic retrieval models — query expansion and document a priori models. Firstly, we will only consider monolingual scenarios, in which documents and queries have the same language.

Cross-lingual IR: We present different approaches to CLIR. This including translation based and concept based approaches. We also discuss the advantages and disadvantages of the different approaches in respect to specific application scenarios.

Multilingual IR: We introduce MLIR and distinguish it to the related problem of CLIR. In multilingual scenarios, an additional problem is often to detect the right language of documents, which we introduce as the language identification problem. Another challenge for MLIR is the organization and construction of indexes. We introduce different approaches to define multilingual indexes and define aggregated retrieval models that allow to build rankings of multilingual documents.

Evaluation: We present the most common methodologies for the evaluation of IR systems. This includes experimental setup, relevance assessments and evaluation measures. We also give an overview of established datasets.

Tools, Software and Resources: Finally, we present selected tools that can be used to implement the different steps in the IR pipeline.

Algorithm 1 The preprocessing pipeline of document d that results in a set of tokens T.

```
d ← INPUT
T ← ∅
[c_1, c_2, ...] ← character-stream(d)
B ← tokenize([c_1, c_2, ...])
while B ≠ ∅ do
  t ← POLL(B)
  if is-compound(t) then
    B ← B ∪ compound-split(t)
  end if
  if not is-stop-word(t) then
    t = normalize(t)
    T ← T ∪ {t}
  end if
end while
return T
```

II.1 Document Preprocessing

In this section, we cover the preprocessing of documents. Preprocessing takes a set of raw documents as input and produces as set of *tokens* as output. Tokens are specific occurrences of terms[2] in a document which represent the *smallest unit of meaning*. This output defines the *vocabulary* that can be used to index a collection of documents as described in Section II.2.

As in most extant IR models and systems, we make the simplifying assumption that the order of tokens in a document is irrelevant. *Phrase indexes* or *positional indexes* are examples of approaches making use of the order of tokens (see Manning et al. [2008] for an introduction).

Depending on language, script and other factors, the process for identifying terms can differ substantially. For Western European languages, terms used in IR systems are often defined by the words of these languages. However, terms can also be defined as a fixed number of characters in sequence, which is often the case in IR systems applied to Asian languages. In Chinese for example — where words are not separated by whitespaces — it is common to use character sequences instead of words as tokens. This avoids the problem of detecting word borders.

In the following sections, we introduce common techniques used for document preprocessing. In our discussion, we will mainly emphasize the differences in preprocessing for different languages and scripts. This includes in particular a discussion of the differences with respect to document syntax, encoding, tokenization and normalization of tokens. The complete preprocessing pipeline is illustrated in Algo-

[2]Terms are also referred to as "types" in literature.

rithm 1. This pipeline shows the dependencies of the different preprocessing steps that will be described in the following.

II.1.1 Document Syntax and Encoding

The first step in the preprocessing pipeline is to identify documents in the given data stream [Manning et al., 2008]. In many cases this is a straightforward task as one can assume that one file or web site corresponds to exactly one document. However, there are other scenarios with files containing several documents (for example XML retrieval) or documents spread over several files (for example Web pages). The definition of what exactly constitutes a document is thus essentially a design choice of the developer and depends on the search task, which is defined by the kind of information items that should be retrieved.

The next step is to transform documents into character streams representing the content of documents. The goal of this step is to get a unified representation with respect to encoding, script and direction of script. Two documents with the same content in the same language should have thus identical character streams after this step. The following challenges have to be addressed:

Document Syntax. The content of documents is usually encoded in a syntax specific for the given *file type*. Based on the file type specification, the textual content of a document has to be extracted prior to indexing, avoiding in particular that vocabulary elements containing formatting instructions or metadata information are indexed.

Examples for file types which require content extraction are PDF files or Web pages. For both cases, many libraries are available that parse PDF or HTML files and extract the textual content.

In many scenarios, only parts of the textual content contribute to the semantic content of an individual document. Other parts might be equal across documents — for example header or footer parts. In these cases, indexing the complete text also introduces irrelevant content. For specific document formats, extractors based on the structure of documents are usually applied to identify the relevant text parts. In the case of Web pages for example, the extraction depends on the individual layout of the site. While elements from the header like title or keywords might describe content and should be extracted, the top bar or menus — which are identical on all web pages in the collection — should be ignored.

Encoding and Script. Encoding refers to the representation of letters through a number of bytes in a computer system. Historically, the ASCII character encoding scheme was widely used to encode English documents. As this scheme is mainly limited to the Latin alphabet, it cannot be used for scripts having different letters. As an encoding scheme supporting most common languages, Unicode [Consortium, 2009] established itself as the *de facto* standard for internationalized applications.

The unique number of every character across all languages ensures high portability across platforms and avoids errors introduced by conversion. Unicode also supports right-to-left scripts and can be used to encode for example Arabic or Hebrew. As most operating systems and also most current programming languages support Unicode, it is highly recommended to use this character encoding as default.[3]

For IR systems, it is essential that queries and documents are represented in the same script. At some level, retrieval boils down to matching characters, which will not be successful when query and document scripts are not compatible. Korean is an example for a language that has different common scripts, namely Hangul and Hanja. The process of mapping text into another script is defined as *transliteration*. This must not be confused with *translation*, as the language is not changed. Transliteration attempts to mimic the sound of the word in the original language by using the spelling of a different language. It is therefore a phonetic transformation that is typically reversible.

For preprocessing of datasets containing documents in heterogeneous scripts, *romanization* is a common technique used to get a unified representation. Romanization is defined as the transliteration of any script to the Latin (Roman) alphabet. For example, this allows to represent Chinese text using letters from the Latin alphabet. For retrieval systems, this is especially useful when searching for common names. Applying romanization, common names used in documents of different languages and scripts will be mapped to the same character sequence in most cases. As part of the United Nations Group of Experts on Geographical Names (UNGEGN), the Working Group on Romanization Systems provides romanization resources for various languages. The motivation of this working group is to introduce unique representations of geographic names. However, the resources provided can be used for romanization of any text.

Direction of Script. As scripts are used to record spoken language, there is a natural order of words and characters defined by their order in the speech stream [Manning et al., 2008]. Usually byte representations of text also reflect this natural order. The actual direction of script is handled by the visualization layer in applications, which is part of the user interface. The main problems are typically documents containing text in different languages with different directions of script. An example are Arabic texts with English common names. As we focus on the core functionality and models of multilingual IR in this chapter, we will not discuss these difficulties any further. We only operate on the data level of documents, which is usually independent of the direction of the script. When designing user interfaces, this is a more important issue.

[3]Technically, Unicode is a mapping from characters to code points. The set of Unicode encodings includes UTF-8 and UTF-16.

II.1.2 Tokenization

Tokenization is defined as the process of splitting a character stream into tokens. Tokens are instances of terms and correspond to the smallest indexation unit. The set of all terms is typically called the *vocabulary*. In the following, we will introduce three common types of vocabularies that require different tokenization approaches. The choice of vocabulary is an important design choice for any IR system.

In order to illustrate the different tokenization approaches we will use the following sentence as running example:

```
It is a sunny day in Karlsruhe.
```

Word Segmentation. The most common approach to tokenization is splitting text at word borders. Thus, tokens refer to words of a language and the vocabulary is equivalent to a lexicon (including morphemes).

For languages that use whitespaces to separate words, this is a successful approach used in most IR systems. Whitespaces and punctuations are thereby used as clues for splitting the text into tokens. Examples for such languages are Western European languages. The problem of this approach is that simply splitting text at all whitespaces and punctuations will also split parts of the text that should be represented by a single token. Examples of such error sources are hyphens (*co-education*), white spaces in proper nouns (*New York*), dates (*April 28, 2010*) or phone numbers [Manning et al., 2008]. In many cases heuristics are used to decide whether to split or not. Classifiers can also be trained on this decision. For our running example, splitting using white spaces results in the following tokens:

```
[It], [is], [a], [sunny], [day], [in], [Karlsruhe]
```

Tokenization at word borders is a much harder problem for scripts without whitespaces — such as Chinese. Approaches can be classified into two classes: lexical and linguistic. Lexical approaches match terms from a lexicon to the token stream in order to get a complete coverage. Usually this matching is not deterministic. To get reasonable accurate matchings, heuristics can be applied — such as always preferring to match longer terms. A problem for such approaches are *unknown terms* which are not in the lexicon and will not be matched but should be detected as such. Linguistic approaches make use of background knowledge consisting of already tokenized text. Using statistical measures based on the frequency of tokens, the goal is to find the most probable segmentation of the current text. *Hidden Markov Models* can be used to compute this in an efficient way [Zhang et al., 2003]. Machine learning techniques like *Conditional Random Fields* have also been successfully applied to this problem [Peng et al., 2004]. As all approaches never achieve a perfect segmentation, wrong tokens will be used for indexing and search and will therefore degrade the retrieval performance.

Phrase Indices. Phrase indices are based on word segmentation. Tokens are thereby defined not as single words but as sequences of words. Phrase indices are also known as *n-gram models*, with n defining the number of words in each token. The character stream which has been already split into words is mapped to tokens by moving a window of n words iteratively over the text. These tokens preserve the context of words. However, this comes at the cost of a very huge vocabulary. Another problem for search is the sparseness of terms, leading to the problem that many terms in queries will not be present in the collection at all. In order to circumvent this problem, phrase indices can be used on top of a retrieval approach based on single word segmentation. For our running example, a 3-gram model results in the following tokens:

```
[It is a], [is a sunny], [a sunny day], ...
```

Character n-gram Models. Character n-gram models use sequences of n characters as tokens. Token streams are obtained by moving a window of n characters over the text. In this case terms do not correspond to words. The vocabulary is defined as the set of sequences having n characters, including white spaces and punctuation. Term lengths of 4 or 5 have been shown to be reasonable. For our running example a character 4-gram model results in the following tokens:

```
[_It_], [It_i], [t_is], [_is_], [is_a], [s_a_], ...
```

This approach can be applied to any character stream and does not depend on word border clues such as whitespaces. It can therefore be used to tokenize text of any script. As no segmentation is needed, this also avoids errors introduced by word segmentation. In the literature, this approach has also been proven to be superior to word based segmentation in several scenarios [McNamee and Mayfield, 2004]. It has also been applied to the problem of spelling correction [Manning et al., 2008].

In the context of multilingual retrieval, character n-gram tokenization can only be used if no mapping of terms into different languages is needed. Terms do not correspond to words and thus can not be mapped or translated across languages. Another drawback using character n-grams is the more difficult visualization of search results. As only n-grams are matched, it is not possible to highlight matching words in the search results.

II.1.3 Normalization

The goal of normalization is to map different tokens describing the same concept to the same terms. An English example is the mapping of plural forms to their singular forms, like *cars* to *car*. Normalization can be defined as building equivalence classes of terms. For each equivalence class, a representative term is selected, which is then used to replace the occurrences of all other terms in this class. Often, the most frequent term in each class is used as such representative. In a search scenario, normalization can be used to increase the amount of relevant documents retrieved

and thus also increase the recall of the system. The same normalization methods have to be applied to the collection before indexing and to the query before search. This ensures that all tokens are mapped to equivalent terms — which is crucial for matching queries to documents.

The approach to normalization differs between languages. For languages having complex morphology, it is a common approach to map (compound) terms to their lemma(s). Examples are Roman and Germanic languages. There are two main approaches to this problem. Firstly, *lemmatizers* use lexical information to map terms to lemmas. For this approach rich linguistic resources are required. Secondly, *stemmers* use a set of simple rules to map terms to stems [Manning et al., 2008]. For the plural example, the rule of deleting trailing *s* would map both terms to the same equivalence class. Stemmers do not require rich language resources. The drawback is that terms are not mapped to lemmas but to stems, which do not necessarily correspond to a word. In many cases terms describing different concepts might also be mapped to the same stem. For example, the terms *organize*, *organizing* and *organization* would be mapped to *organ*, which makes it impossible to distinguish these terms in the index. In contrast, a lemmatizer would correctly normalize *organize* and *organizing* to the lemma *organize* without changing the term *organization*.

Normalization might also be useful for scripts using diacritics. An example of the usage of diacritics in French are the characters “é” or “è”. If the usage of diacritics is not consistent, it is useful to delete them in the normalization step. For example, if users do not specify diacritics in queries, normalization should also delete them before indexing. Simple rule-based approaches are normally applied to remove diacritics.

For fusional languages (for example German, Dutch, Italian), *compound splitting* is another form of normalization. Applied to such languages, compound terms are typically split into the composing lemmas in order to increase recall. The problem of compound splitting is quite similar to the problem of word segmentation in Asian languages as described above. Lexical approaches use a lexicon to match terms in compounds. Linguistic approaches additionally use background knowledge. Many approaches compare the frequency of the compound itself and the frequency of its constituent terms to decide whether to split or not. When applying compound splitting, usually both compounds and split components are added to the token stream. In the search process this still allows the matching of compounds.

Removal of *stop words* is a normalization step that deletes frequent terms from the token stream. The frequency of these terms is thereby counted in the whole document corpus. As a consequence, these most frequent corpus terms occur in almost all documents and are therefore not useful to discriminate between relevant and non-relevant documents. Stop words are usually articles, prepositions or conjunctions. For many languages, compiled lists of stop words exist which can be used to match and filter tokens. For example, an English stop word list may contain the following terms:

```
a, in, is, it, the, this, to, when, ...
```

Coming back to our running example, stemming and stop word removal would result in the following tokens:

```
[sunni], [day], [karlsruh]
```

Given some simple rules, the trailing "y" of `sunny` is substituted by "i" and the trailing "e" of `Karlsruhe` is omitted.

II.1.4 Reference to this Thesis

The experiments presented in this theses are only based on documents and queries in Roman and Germanic languages. As all of these languages are based on the Latin alphabet, we use word segmentation based on whitespaces throughout the experiments. As normalization step, stemming and stop word removal are applied to documents and queries.

Other techniques — such as transliteration or character n-gram models — will not be used in our experiments. However, our generic approaches could be applied to other languages like Chinese or Korean which require other preprocessing steps. When applied consistently to documents, queries and background knowledge, the variation of preprocessing would most probably not affect the effectiveness of our proposed retrieval models.

II.2 Monolingual IR

Most approaches to MLIR are either directly based on monolingual IR techniques or make at least use of standard IR models. MLIR can be seen as the problem of aggregating the results of IR systems in different languages. Apart from aggregation, language specific preprocessing of queries is needed, in particular translation which will be covered in Section II.4. In general, MLIR is based on the same index structures and relies on similar document and retrieval models as known from monolingual IR. In this chapter, we therefore give a short overview of monolingual IR, including document representation, index structures, retrieval models and document *a priori* models. We focus on those aspects of IR which will also be relevant for CLIR or MLIR. For more details concerning monolingual IR we refer to Manning et al. [2008] or Baeza-Yates and Ribeiro-Neto [1999].

II.2.1 Document Representation

In Section II.1, we described the preprocessing of documents. This results in *token stream* representations of documents. The tokens are instances of terms, which are defined by words, stems or lemmas of words or character n-grams. The IR models presented in this chapter are independent of the used vocabulary and can be applied

to any term model. For the sake of presentation, we will make the simplifying assumption that terms correspond to words in spoken language throughout this chapter, as this yields the most intuitive vocabulary for humans.

Most current retrieval approaches use document models based on the *independence assumption* of terms. This means that occurrences of terms in documents are assumed to be independent of the occurrences of other terms in the same document. While this is certainly an overly simplistic assumption, retrieval models based on this assumption achieve reasonable results with current IR technology.

Given the independence assumption, documents can be represented using the *vector space model*. The vector space is spanned by the vocabulary in such a way that each dimension corresponds to a specific term. Documents are represented as vectors by a mapping function f which maps token streams of documents d to term vectors $\vec{d}$. Different functions f are used in literature, the most prominent being:

Boolean Document Model: The value of a dimension corresponding to a specific term is set to 1 if the term occurs at least once in the document, otherwise to 0.

TF Document Model: The value of each dimensions depends on the number of occurrences of terms in the document token stream, defined as *term frequency*. The term frequency can be directly used as value in the term vector. Variants are for example the normalization of the term frequency by document length.

TF.IDF Document Model: These models additionally multiply term frequency values by the *inverse document frequency* of terms. The document frequency of a term is the number of documents in the collection containing this term. The inverse document frequency therefore puts more weight on seldom terms and less weight on frequent terms which do not discriminate well between documents in the collection. In most cases, the logarithm of the inverse document frequency is used in TF.IDF models.

Given a collection of documents, the document term vectors can be aligned to form the *term-document matrix*. This matrix is spanned by terms as rows and documents as columns. We will illustrate the different document representations using the following documents:

```
Doc1: It is a sunny day in Karlsruhe.
Doc2: It rains and rains and rains the whole day.
```

For the different documents models discussed, this results in the following term-document matrices:

	Boolean		*TF*		*TF.IDF*	
Term	Doc1	Doc2	Doc1	Doc2	Doc1	Doc2
sunny	1	0	1	0	$1 \log 2/1 = 0.7$	0.0
day	1	1	1	1	$1 \log 2/2 = 0.0$	$1 \log 2/2 = 0.0$
Karlsruhe	1	0	1	0	$1 \log 2/1 = 0.7$	0.0
rains	0	1	0	3	0.0	$3 \log 2/1 = 2.1$

II.2.2 Index Structures

An important aspect of IR is time performance. Users expect retrieval results in almost real time and delays of only one second might be perceived as a slow response. The simplistic approach to scan through all documents given a query does obviously not scale to large collections. The high time performance of current retrieval systems is achieved by using an *inverted index*. The idea is to store for each term the information in which documents it occurs. This relation from terms to documents is called *posting list*, a detailed example can be found in Manning et al. [2008]. During retrieval, only posting lists of query terms have to be processed. As queries usually consist of only few terms, the scores can be computed with low average time complexity.

For the example documents presented above, we get the following posting lists:

```
sunny     -> doc1(1x)
day       -> doc1(1x), doc2(1x)
Karlsruhe -> doc1(1x)
rains     -> doc2(3x)
```

A remaining bottleneck using inverted indexes is memory consumption. Loading of posting lists from storage to main memory is the slowest part and should be avoided. Heuristics are therefore needed to decide which posting lists should be kept in memory and which should be replaced. General approaches to reduce memory usage — for example by compression or by usage of suffix trees — are described in [Baeza-Yates and Ribeiro-Neto, 1999]. For very large corpora, distributed indexing can be applied. Posting lists are distributed to several servers. Each server therefore indexes the posting lists of a subset of the vocabulary.

In order to reduce the time complexity of retrieval, *inexact retrieval models* — also known as *top-k models* — can be applied. These models determine documents that are most likely to be relevant without processing all matching documents. Using these methods, retrieval time can be reduced without getting significant losses in retrieval performance [Anh et al., 2001].

II.2.3 Retrieval Models

Retrieval models are used to estimate relevance of documents to queries. Different theoretical models have been used to define these relevance functions. In the following, we will describe three main families of retrieval models: *boolean models*, *vector space models* and *probabilistic models*. Depending on the retrieval model, queries are represented in different ways. Boolean queries used for boolean models are modeled as a binary term vector. As defined above, the order of query terms is lost in this representation, as only the presence or absence of terms is captured. For vector space and probabilistic models, queries are represented in a real-valued vector space and scores for each query term are accumulated [Manning et al., 2008].

Boolean Models. Boolean Models have been the first retrieval models used in the beginning of IR. In the case of the Boolean retrieval model, relevance is binary and is computed by matching binary vectors representing term occurrence in the query to binary document vectors representing term occurrence. As current vector space or probabilistic models outperform boolean models, we will not consider boolean models in this chapter but focus on the other more successful models. The interested reader is referred to [Manning et al., 2008] for details.

Vector Space Models. Vector space models are based on vector space representations of documents. As described above, this vector space is spanned by the vocabulary and entries in the term-document matrix are usually defined by term frequencies. There are different models to assess the relevance of documents to a given query:

- *Accumulative Model:* The retrieval function computes scores for each query term. The query term scores are summed up per document to get a final accumulated score for each document. Functions computing scores for a single query term t are based on the following measures:
  - $\text{tf}_d(t)$. Term frequency in the document.
  - $|d|$. Length of the document.
  - $\text{df}(t)$. Document frequency of the query term.
  - $\text{tf}_D(t)$. Number of tokens of the query term in the whole collection.
  - $|D|$. Number of documents in the collection.

  For example, the accumulated score of a simple retrieval model based on term frequency and inverse document frequency is computed as follows:

$$\text{score}(q,d) = \sum_{t \in q} \text{tf}_d(t) \log \frac{|D|}{\text{df}(t)}$$

- *Geometric Model:* The vector space representation of the query q can be interpreted as term vector $\vec{q}$. In this case, geometric similarity measures in the term vector space can be used as retrieval models [Manning et al., 2008]. For example the *cosine similarity* has been applied successfully in retrieval scenarios:

$$\text{score}(q,d) = \text{cosine}(\vec{q},\vec{d}) = \frac{\langle \vec{q},\vec{d} \rangle}{\| \vec{q} \| \| \vec{d} \|}$$

Probabilistic Models. In probabilistic retrieval models, the basic idea is to estimate the likelihood that documents are relevant to a given query. Relevance is thereby modeled as a random variable R taking values $\{1,0\}$. A document d is relevant for a given query q, iff $P(R=1|d,q) > P(R=0|d,q)$ [Manning et al., 2008,

p. 203]. It has been shown that, given a binary loss function and the most accurate estimation of all probabilities based on all available information, these models achieve optimal performance [van Rijsbergen, 1979]. However, in practice it is not possible to get accurate estimations. Probabilistic models have also been used to justify design choices in heuristic functions used in vector space models, for example the use of the inverted document frequency (see [Manning et al., 2008] for more details).

The BM25 model [Robertson and Walker, 1994] is an example of a probabilistic retrieval model that has been proven to be very successful in practice. The scoring function is defined as follows:

$$\begin{aligned} \text{score}(q,d) &= \sum_{t \in q} \text{idf}(t) \frac{\text{tf}_d(t)}{k_1 \left((1-b) + b \frac{|d|}{\frac{\sum_{d'} |d'|}{|D|}} \right) + \text{tf}_d(t)} \\ \text{idf}(t) &= \log \frac{|D| - \text{df}(t) + 0.5}{\text{df}(t) + 0.5} \end{aligned}$$

Common values for the parameters of this model are $k_1 = 2$ and $b = 0.75$, but they should be adjusted to the search task and dataset.

Language Models. In recent years, language models have established themselves as powerful alternative retrieval models. Language models are a subclass of probabilistic models. Documents, queries or whole collections are represented by generative models. These models are represented by probability distributions over terms, for example the probability that a document, query or collection generate a certain term [Ponte and Croft, 1998].

Maximum likelihood estimation is often used to define document models. The probability of a term t being generated by document d is then defined as:

$$P(t|d) = \frac{\text{tf}_d(t)}{|d|}$$

In information retrieval, language models are used to estimate the probability $P(d|q)$ which is then interpreted as relevance score. Using *Bayes' Theorem*, this can be transformed to:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

As $P(q)$ is constant for a query and $P(d)$ can be assumed to be uniform, ranking of documents is based on the value of $P(q|d)$. When modeling queries as set of independent terms, this probability can be estimated using document language models:

$$P(q|d) = \prod_{t \in q} P(t|d)$$

As this score will be zero for all documents not containing all query terms, *smoothing* is often applied. Using background knowledge, *a priori* probabilities of terms $P(t)$ are estimated and a mixture model is used for retrieval:

$$P(q|d) = \prod_{t \in q} (1 - \alpha) P(t|d) + \alpha P(t)$$

with α being the smoothing weight.

Often, the whole collection is used as background knowledge and the *a priori* probability is estimated by the language model of the collection:

$$P(t) = \frac{\sum_{d \in D} \mathrm{tf}_d(t)}{\sum_{d \in D} |d|}$$

II.2.4 Query Expansion

Query expansion is an established technique to improve retrieval performance, which is of special interest in the context of cross-lingual and multilingual IR. The query is expanded by additional terms that further characterize the information need. The goal is to match more relevant documents which contain relevant content but use other terms to describe it.

Expanded queries can be used in all retrieval models presented above. Usually, expanded query terms are given less weight than the original query terms. The weight depends on the confidence of each expanded term and the overall weight put into query expansion. Using probabilistic retrieval models, query expansion can be used to improve the estimation of probabilities, for example the estimation of the query language model. We distinguish two different sources for expansion terms:

Background Knowledge. Additional knowledge sources are exploited to find expansion terms for a given query. An example is the usage of a thesaurus to expand the query with synonyms of the query terms [Voorhees, 1994; Mandala et al., 1999]. For CLIR or MLIR, a special case of query expansion is the translation of the query. In this case the query is expanded using the terms of its translation into different languages.

Relevance Feedback. Using *relevance feedback* for query expansion is a two step retrieval process. First, the original query is matched to the document collection. Then, relevance assessments are used to identify the relevant documents in the retrieval results. Using an expansion model based on term frequency and document frequency in this set of *expansion documents*, promising terms are identified and used in a second retrieval step for query expansion.

The selection of relevant documents in the first step can be either manual or automatic. In the first case, the user selects relevant documents manually out of the retrieval results of the first step. In the case of so called *Pseudo Relevance Feedback*

(PRF), the top k documents of the first retrieval step are assumed to be relevant [Xu and Croft, 1996]. This enables to implement automatic query expansion without user interaction. For this reason, PRF is often referred to as *blind relevance feedback*.

II.2.5 Document a priori Models

In all retrieval models presented above, the *a priori* probability of documents is assumed to be uniform, *i.e.* the probability of retrieving documents independently of a specific query is the same for all documents. However, in many scenarios this assumption does not hold. For example, documents have different perceived quality and popularity. Such factors could definitely influence the *a priori* probability of a document in the sense that very popular or high-quality documents should intuitively have a higher likelihood of being relevant.

For the different types of retrieval models, there are different approaches to integrate document priors. When using vector space models, these *a priori* probabilities can be multiplied with the IR score of each document [Baeza-Yates and Ribeiro-Neto, 1999]. Another option is the linear combination of scores and *a priori* probabilities. Weights in this linear combination have to be optimized for the special application scenario. For probabilistic and language models, the estimation of document priors $P(d)$ is required as part of the retrieval model. In the standard case without background knowledge, document priors are assumed to have equal distribution over all documents. However, if document *a priori* models are available, they can be integrated directly into the retrieval model by replacing the uniform model used before.

The way document priors are modeled clearly depends on the target application. In web search for example, the web graph consisting of pages and hyperlinks can be exploited to compute authority measures, which can be used as document priors. Pagerank [Brin and Page, 1998] and HITS [Kleinberg, 1999] are established algorithms to compute authority in Web graphs. Another example is search in community portals. Ratings of users, usage patterns or other evidence can be used to compute document priors [Agichtein et al., 2008].

II.2.6 Reference to this Thesis

In Chapter IV, we present a concept-based retrieval model. This model makes use of the vector space model to relate documents to concepts. Further, the retrieval model in the concept space is inspired by the geometric model presented above.

The theory of language models is used in Chapter V to define new generative models that support the aggregation of different sources of evidence in the retrieval process. Standard approaches based on probabilistic retrieval models are thereby used as baseline in the experiments.

II.3 Cross-lingual IR

Cross-lingual IR is the task of retrieving documents relevant to a given query in some language (query language) from a collection D of documents in some other language (collection language) (see Definition I.7). Hereby, D is a monolingual collection, *i.e.* all documents in D have the same language.

Essentially, we can distinguish between two different paradigms of CLIR. On the one hand, we have translation-based approaches that translate queries and/or documents into the language supported by the retrieval system. Such approaches reduce the task of cross-language retrieval to a standard monolingual IR task to which standard retrieval techniques can be applied. On the other hand, there are also approaches that map both documents and queries into an interlingual (concept) space. The relevance functions are then defined on the basis of this interlingual space. We discuss these different approaches below.

II.3.1 Translation-based Approaches

Translation-based approaches translate the query and/or the document collection into some language supported by the retrieval system. Translation-based approaches differ in the choice of translation techniques as well as in the choice of whether only the query, the document collection or both are translated. We will describe several alternative choices for the latter below. Further, translations can be either obtained by involving manual translators or through the application of Machine Translation (MT) techniques. We also discuss this in more detail below.

Translating Queries. The default strategy for CLIR is the translation of the query into the language of the document collection. This effectively reduces the problem of CLIR to monolingual IR. In what follows, we list some of the advantages (PRO) and disadvantages (CON) of such an approach:

PRO

- Only the query has to be translated, which is usually a short text.
- The index can be used to evaluate queries in arbitrary languages (under the condition that they can be translated into the language of the collection / index).

CON

- An online query translation is needed. As the response time of the retrieval system is the sum of the translation time and the retrieval time, an efficient MT system is needed in order to maintain system performance at reasonable levels.

- The accuracy of the retrieval system is partially dependent on the quality of the MT system used.

Translating Documents. A further strategy is to translate the entire document collection into the query language and create an inverted index for the query language. This might be useful in search scenarios having a fixed query language, for example in portals that have only users of one language. In the following, we also provide a summary of advantages and disadvantages of such an approach:

PRO

- The translation is part of the preprocessing as indices will be based on the translated documents. Thus, there is (almost) no temporal constraint on the translation step — such that one can resort to manual translation if needed for quality reasons.

CON

- The query language has to be known and fixed in advance. As the index is specific for this language, queries in other languages are not supported.
- The entire collection has to be translated, which might be costly.

Pivot Language. As a combination of the first two approaches, both queries and documents can be translated into a so-called *pivot language*. The pivot language is either a natural or artificial language for which translation systems are available from many languages. English is most often used as such pivot language due to the large amount of available translation systems. As no direct translation from query language to document language is needed, the pivot language approach is useful if no language resources supporting this translation are available.

Using a pivot language reduces CLIR to the problem of standard monolingual IR as an existing IR system in the pivot language can be applied to any pairs of query and document languages. However, the performance depends on an adequate translation for both the query language and the collection language into the pivot language. Advantages and disadvantages here can be summarized as follows:

PRO

- Translation systems to a pivot language can be used for CLIR between languages for which direct translation is not available.
- Existing IR systems in the pivot language can be used for CLIR for any pair of query and document language.

CON

- Online translation of the query as well as offline translation of documents (as part of document preprocessing) are required.

Query Expansion. Query expansion techniques that add additional query terms to the original query can also be applied in CLIR settings in the following ways:

Pre-translation expansion expands the query before it is translated. The expanded query is then processed by the translation system. This has the advantage that more context by the additional query terms is given as input to the translation process. In CLIR settings, this was shown to improve precision of the retrieval results [Ballesteros and Croft, 1997].

Post-translation expansion is equivalent to query expansion used in monolingual IR. In a CLIR setting, it has been shown that post-translation expansion can even alleviate translation errors as wrong translations can be spotted by using local analysis of the results of the query (*e.g.* using PRF) [Ballesteros and Croft, 1997].

II.3.2 Machine Translation

As described above, a translation step is necessary for translation-based CLIR. Either the query or documents need to be translated before queries can be evaluated using the (language-specific) inverted index. Manual translation — for example by professional translators — typically incurs high costs. The manual translation of documents does not scale to large corpora and it is not possible to have real-time translation of queries, a crucial requirement in retrieval systems that need response times in fractions of a second (for example in web search). This clearly motivates the use of MT for CLIR.

In this chapter, we present the two main approaches to MT used in CLIR systems: *dictionary-based translation* and *Statistical Machine Translation*.

Dictionary-based Translation. A straight-forward approach to query translation is the use of bi-lingual dictionaries for term-by-term translation. There are different strategies to cope with alternative translations of terms, ranging from choosing the most common translation to taking into account all possible translations. Interestingly, Oard [1998] has shown that there are no significant differences between the different strategies in a CLIR setting. When using all alternative translations, query terms are usually weighted by their translation probability.

Ballesteros and Croft [1997] argue that post-translation expansion can be used to minimize translation errors in dictionary-based query translation. They have shown that using Pseudo Relevance Feedback for query expansion removes extraneous terms introduced by the translation and therefore improves the retrieval performance.

Statistical Machine Translation. In contrast to dictionary-based translation, Statistical Machine Translation (SMT) aims at translating whole sentences. Thus, in principle SMT can be applied both in the case of query or document translation.

Most current SMT systems are based on the IBM Models introduced by Brown et al. [Brown et al., 1993]. These models are iteratively induced for language pairs on the basis of a training corpus in which sentences are aligned across languages. In two subsequent steps, the term alignment of translated sentences and the translation model of terms are optimized. The final model then is a product of the iterative optimization of these two steps. These models can be further improved by additionally translating phrases. In this case, not only alignment and translation of single terms but also of phrases like *New York* are learned and applied. Using additional background knowledge can also improve translation, for example by including language models derived from large monolingual training corpora.

The drawbacks of applying SMT systems to translate the query on the fly is the potentially longer execution time of the retrieval step and the requirement of a training corpus. The execution time bottleneck seems less problematic given the continuous advances in computer hardware and the fact that providers of online translation systems build on a large and distributed computer infrastructure. Indeed, recent systems can already be applied to real time query translation. However, training corpora are still missing for many language pairs [Resnik and Smith, 2003].

II.3.3 Interlingual Document Representations

An alternative to translation-based CLIR are interlingual document representations. The essential idea is that both query and documents are mapped to an interlingual *concept space*. In contrast to term-based representations of documents, concepts represent *units of thought* and are thus assumed to be language-independent. Language-specific mapping functions are however needed in order to map documents into the interlingual concept space. Such a mapping might for instance rely on a quantification of the degree of association for terms in different languages (and by aggregation also for a document) to the given set of interlingual concepts.

By mapping queries to the same concept space as documents, IR can be reduced to the comparison of query and document concept vectors. This enables the application of standard similarity measures to compute a ranking, such as the cosine of the angle enclosed by the two vectors representing the query and the document. In the following, we present two approaches to interlingual concept spaces that have been applied to CLIR.

Latent Semantic Indexing. In the monolingual case, Latent Semantic Indexing (LSI) is used to identify latent topics in a text corpus. These topics, which correspond to concepts as described above, are extracted by exploiting co-occurrences of terms in documents. This is achieved by Singular Value Decomposition of the term-document matrix [Deerwester et al., 1990]. The latent topics then correspond to the

eigenvectors having the largest singular values. This also results in a mapping function from term vectors to *topic vectors*. LSI was originally used for dimensionality reduction of text representation and for improved retrieval of synonyms or similar terms. By using parallel training corpora, LSI can also been applied to CLIR [Dumais et al., 1997]. In this case the extracted topics span terms of different languages and the mapping function maps documents of all languages to the latent topic space. More details of LSI will be given in Chapter III.

Explicit Semantic Analysis. Recently, Explicit Semantic Analysis (ESA) has been proposed as an alternative concept based retrieval model [Gabrilovich and Markovitch, 2007]. Concepts are explicit and defined with respect to some external knowledge source. Textual descriptions of each concept are used to map documents into the concept space. Examples of such resources of concepts and their descriptions that have been used for ESA are Wikipedia and Wiktionary. ESA can be applied to CLIR if textual descriptions of concepts are available in all languages supported by the retrieval system. When using Wikipedia as multilingual knowledge resource, cross-language links can be used to build multilingual concept definitions, which will be introduced in Chapter IV.

II.3.4 Reference to this Thesis

We will introduce Cross-lingual ESA as a CLIR system in Chapter IV. In our experiments we compare the retrieval performance of CL-ESA, LSI and Latent Dirichlet Allocation (LDA) on established datasets. Further we present an aggregated model that combines translation based approaches with CL-ESA. This model was evaluated by our participation in an international MLIR challenge.

The language models presented in Chapter V are instances of translation based approaches. In the retrieval process, queries are translated to all the different document languages.

II.4 Multilingual IR

In contrast to CLIR, Multilingual IR (MLIR) considers corpora containing documents written in different languages. Given the documents in languages $l_1, \ldots, l_n$ of collection D, the task is defined as the retrieval of a ranked list of relevant documents for a query q in the query language l_q. These relevant documents may thereby be distributed over all languages $l_1, \ldots, l_n$ (see Definition I.8).

In general, MLIR systems are based on similar techniques as CLIR systems and essentially the same translation approaches can be applied. However, the multilingual scenario requires a different index organization and relevance computation strategies compared to both monolingual and cross-lingual retrieval. In the following, we briefly describe different strategies ranging from unified indices to multiple

language-specific indices. If the language of the documents is not known *a priori*, language identification is required as part of the preprocessing.

II.4.1 Language Identification

Language identification is defined as the problem of labeling documents with the language in which the content of the document is expressed. In the following, we assume that documents are monolingual, *i.e.* they contain text in one language. The more complex case of mixed documents will be briefly touched upon at the end of this section.

The problem of language identification can be reduced to a standard classification problem with discrete classes. The target classes are given by a set of languages and the task is to classify documents into one of these classes representing each of the relevant languages. Given monolingual training corpora for each language, supervised Machine Learning (ML) approaches can be applied to this task. The most successful reported classification method is based on character n-gram representations of documents. Cavnar and Trenkle [1994] present language identification results with 99% of precision using a set of 14 languages. They build term vectors for each document by extracting character n-grams for $n = 1 \ldots 5$. An important aspect here is that the classifiers can be trained on monolingual input for each of the languages, so that an aligned dataset is not necessary. Thus, the approach is applicable in principle to any set of languages. Further, the method requires no preprocessing as character n-grams are based on the character streams without word splitting. It has been demonstrated that the accuracy of this language identification method is dependent on document length as longer documents provide more evidence for the language they are written in. The results of Cavnar and Trenkle [1994] show that precision of 99% or more can be expected for documents having more than 300 characters.

Applying the proposed classifier on mixed documents having content in multiple languages results in unpredictable classification. The language-specific distribution of terms or n-grams (that is exploited by the classifier) is lost as the characteristics of the individual languages overlay each other. These documents have to be split into their monolingual components beforehand. As the splitting is done on sections, paragraphs or event sentences, this produces shorter documents that are more difficult to classify, thus degrading results.

II.4.2 Index Construction for MLIR

There are two main approaches to index construction for MLIR — differing in whether a single or multiple indices are used. Single index approaches build one index for the documents in the different languages. We distinguish three different techniques for constructing such an index:

Document Translation: By translating all documents into a pivot language, the problem of MLIR can be reduced to CLIR. The single index then contains

all the translated documents.

Language Token Prefixes: Nie [2002] proposes the creation of a unified index by adding language prefixes to all tokens. This ensures that terms having the same character representation in different languages can be distinguished. The lexicon of the unified index consists of terms in all languages. Nie argues that this unified index preserves term distribution measures like term frequency or document length.

Concept Index: As discussed above, language-independent concept indices can also be applied to MLIR. As documents of different languages are mapped to the same interlingual concept space, only a single concept index is needed for the multilingual corpus.

Approaches based on multiple indices build different indices for each language of the corpus. There are two different techniques:

Language-Specific Indices: Each document in a multilingual collection is added to the index for the corresponding language, whereby the language needs to be identified such that the language-specific preprocessing can be applied. For monolingual documents, the set of documents contained in each index are thus disjoint. For mixed documents having content in different languages, only document parts in the language of the index are added. In this case a document can appear in many of the language-specific indices.

Specific Preprocessing: For each language, an index is constructed containing all documents of the corpus. However, preprocessing of these documents is specific to the language associated to each index. For each index, documents are therefore assumed to only have content in the according index language. In the retrieval step, the translated queries are matched to the index associated to the query language.

The indexing approach based on specific preprocessing avoids the problems introduced by language classification, which is mostly relevant for documents having content in several languages. In some cases, text in different languages occurs in the scope of one sentence which can most probably not be detected using language classification. Examples are common names in a foreign language or quotes. Using language specific indexes of all documents with the according preprocessing ensures that each query translation can be matched to the entire corpus. Therefore no text parts are missed due to false language classifications.

II.4.3 Query Translation

The different approaches to index construction require different query translation strategies. For single indices based on document translation or concept indices, we

refer to Section II.3 as the query translation is analogous to query translation used in CLIR.

For all other approaches, queries need to be translated into all document languages. Depending on the index used, these translations are applied in different ways:

Language Token Prefixes: A query for the unified index with language prefixes for each term is build by concatenation of all query translations into a single query and by adding language prefixes to all query tokens. Standard IR retrieval models can then be used to query the unified index.

Multiple indices: When using multiple indices for the different languages, the translation of the query into each language is used to query the index of the corresponding language. This produces different language-specific rankings for each language that have to be combined into an aggregated score determining an aggregated ranking. We discuss some of the most important aggregation models below.

II.4.4 Aggregation Models

Retrieval based on multiple indices requires score aggregation models, as the rankings based on evidence in each language need to be combined to yield a final ranking. Given a set of languages $L = \{l_1, \ldots, l_n\}$, a query q and language-specific scores for each document $\text{score}_l(d, q)$, a straightforward approach is to sum up the scores for all the languages:

$$\text{score}(q, d) = \sum_{l \in L} \text{score}_l(q, d)$$

The aggregated score can then be used to sort all documents and produce an overall ranking of the documents.

The main problem of the above aggregation strategy is the potential incompatibility of the scores. In fact, by simply adding scores it is assumed that the absolute score values express the same relevance level in each ranking. However, for most retrieval models this is not the case. The absolute values of scores depend on collection statistics and term weights, for example the number of documents, number of tokens, average document length or document frequency. For each index, these values differ and therefore the absolute scores are not necessarily comparable.

In order to overcome this problem, typically *normalization* is applied to each ranking before aggregation. A standard approach for MLIR is the Z-Score normalization [Savoy, 2005]. Each ranking is normalized by using statistical measures on its scores, *i.e.* the minimal score, the mean score and the standard deviation. Given training data in the form of queries and relevance judgments for documents, Machine Learning techniques can be used to compute optimal weights by which scores can be combined (see Croft [2002]).

Algorithm 2 Aggregation of multiple rankings $r_1, \ldots, r_n$ based on Z-Score normalization. For ranking r, $r[i]$ defines the score at rank position i, $\text{score}_r(d)$ defines the score of document d. MIN, MEAN and STD-DEVIATION are defined on the set of score values of ranking r.

$R \leftarrow \{r_1, \ldots, r_n\}$
for all $r \in R$ **do** {Normalization}
  $\mu \leftarrow \text{MEAN}(r)$
  $\sigma \leftarrow \text{STD-DEVIATION}(r)$
  $\delta \leftarrow \frac{\mu - \text{MIN}(r)}{\sigma}$
  for $i = 1..|r|$ **do**
    $r[i] \leftarrow \frac{r[i]-\mu}{\sigma} + \delta$
  end for
end for

$r_c \leftarrow []$
for all $d \in D$ **do** {Aggregation}
  $s \leftarrow 0$
  for all $r \in R$ **do**
    $s \leftarrow s + \text{score}_r(d)$
  end for
  $\text{score}_{r_c}(d) \leftarrow s$
end for
$r_c \leftarrow \text{DESCENDING-SORT}(r_c)$
return r_c

A complete aggregation step using Z-Score normalization is presented in Algorithm 2. Given a set of rankings $R = \{r_1, \ldots, r_n\}$, the algorithm computes the combined ranking r_c. In the first step, each ranking r_i is normalized using the minimum value, the mean value and the standard deviation of its values. In the second step, combined scores are computed by summing the scores of each document across all rankings. Finally, the combined ranking is build by re-ordering documents according to descending values of the aggregated scores.

II.4.5 Reference to this Thesis

The scenario of MLIR will be used in Chapter IV and V to evaluate the proposed retrieval models. Standard approaches such as query translation combined with Z-Score normalization will be presented as baseline approaches and will also be integrated in new models. The language model presented in Chapter V uses aggregation not only across languages but also to combine different sources of evidence.

II.5 Evaluation in IR

Ultimately, the goal of any IR system is to satisfy the information needs of its users. Needless to say, user satisfaction is very hard to quantify. Thus, IR systems are typically evaluated building on the notion of *relevance*, where relevance is assessed by a team performing the evaluation of the system rather than by the final user of an IR system. Hereby, one can adopt a binary notion of relevance where documents are relevant to a query or not or even a degree of relevance to a query. The former case is the most frequent one in IR evaluation. Given a specification of which documents are relevant to a certain query and which ones not, the goal of any IR system is to maximize the number of relevant documents returned, while minimizing the amount of non-relevant documents returned. If the IR system produces a ranking of documents, then the goal is clearly to place relevant documents on top and non-relevant ones on the bottom of the ranked list. Several evaluation measures have been proposed to capture these intuitions. In addition, several reference collections with manual relevance judgments have been developed over the years. As results on such datasets are thus reproducible, they allow different system developers to compete with each other and support the process of finding out which retrieval models, preprocessing, indexing strategies, etc. perform best on certain tasks.

In this section, we will first describe the experimental setup that was introduced as the so called *Cranfield paradigm*. Terms defined in the Cranfield experiments — for example *corpus*, *query* or *relevance* — have been used already throughout this chapter. Then, we introduce and motivate different evaluation measures. These measures are based on relevance assessments. We describe manual and automatic approaches to create relevance assessments. Finally, we provide an overview of established datasets that can be used to evaluate CLIR or MLIR systems.

II.5.1 Experimental Setup

The experimental setup used to evaluate IR systems has to ensure that an experiment is reproducible, which is the primary motivation for the development of the Cranfield evaluation paradigm [Cleverdon, 1967]. According to this paradigm, we have to fix a certain corpus as well as a minimum number of so-called *topics* consisting of a textual description of the information need as well as a query to be used as input for the IR system. The systems under evaluation are expected to index the collection and return (ranked) results for each topic (query). In order to reduce the bias towards outliers, a reasonable number of topics needs to be used in order to yield statistically stable results. A number of at least 50 topics is typically recommended.

For each topic, a so-called *gold standard* defines the set of relevant documents in the collection [Manning et al., 2008]. The notion of relevance is hereby typically binary — a document is relevant or not to a given query. Using this gold standard, the IR system can then be evaluated by examining whether the returned documents are relevant to the topic or not, and whether all relevant documents are retrieved. These notions can be quantified by certain evaluation measures which are supposed

to be maximized (see below). As user satisfaction is typically difficult to quantify and such experiments are difficult to reproduce, an evaluation using a gold standard — with defined topics and given relevance assessments — is an interesting and often adopted strategy. We will discuss how the necessary relevance assessments can be obtained in the next section.

II.5.2 Relevance Assessments

Experimentation in IR usually requires so-called *relevance assessments* that are used to create the gold standard. While for smaller collections, for example the original Cranfield corpus, the manual examination of all the documents for each topic by assessors is possible, this is unfeasible for larger document collections [Manning et al., 2008]. Thus, a technique known as *result pooling* is used in order to avoid that assessors have to scan the entire document collection for each topic. The essential idea is that the top ranked documents are *pooled* from a number of IR systems to be evaluated. For each topic, one typically considers the top k documents retrieved by different systems, where k is usually 100 or 1000. As relevance assessments vary between assessors, each document / topic pair is usually judged by several assessors. The final relevance decision as contained in the gold standard is an aggregated value, for example based on majority votes. The measurement of the inter annotator agreement, for example through the kappa statistic [Manning et al., 2008], is an indicator for the validity of an experiment. A low agreement might for instance result from the ambiguous definition of information needs.

By involving several systems in the pooling, one tries to reduce the bias of the relevance judgments towards any single system. Moreover, the test collection should be sufficiently complete so that the relevance assessments can be re-used to test IR techniques or systems that were not present in the initial pool.

For CLIR or MLIR systems, an alternative evaluation method is provided by the so called *mate retrieval* setup. This setup avoids the need to provide relevance judgments by using a parallel or aligned dataset consisting of documents and their translation into all relevant languages. The topics used for evaluation correspond to documents in the corpus. The so-called *mates* of this topic — the equivalents of the document in different languages — are regarded as the only relevant documents, such that the goal of any system is to retrieve exactly these mates. The gold standard can therefore be constructed automatically. The values of evaluation measures are clearly underestimated using this gold standard, as other documents might be relevant as well.

II.5.3 Evaluation Measures

In order to quantify the performance of an IR system on a certain dataset with respect to a given gold standard consisting of relevance judgments, we require the definition of certain evaluation metrics. The most commonly used evaluation measures in IR

	relevant	*non-relevant*
retrieved	TP	FP
non-retrieved	FN	TN

Table II.1: Contingency table of retrieval results for a single query.

are *precision* and *recall*. Precision measures the percentage of the retrieved documents that are actually relevant, and recall measures the percentage of the relevant documents that are actually retrieved.

Computation of these measures can be explained based on the contingency table of retrieval results for a single query as presented in Table II.1. Precision P and recall R are then defined as:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

In order to choose an appropriate evaluation metric, it is crucial to understand how the retrieval system will be used. In some scenarios, users are likely to read all documents, for example in order to compile a report, while in other scenarios, such as ad hoc search in the Web, users are likely to only examine the top-ranked documents. It should be clear from these very extreme examples of usage that the choice of an evaluation metric is not independent of the way the IR system is supposed to be used.

For retrieval systems in which both precision and coverage (*i.e.* returning all relevant documents) is important — which is not necessarily the case for Web Search — a reasonable choice is average precision (AP), which averages the precision at certain positions in the ranking. In particular, these are the positions at which relevant documents are found.

For the document collection $D = \{d_1, \ldots, d_n\}$, ranking $R = \{r_1, \ldots, r_l\}$ with $1 \leq r_i \leq n$ and $r_i \neq r_j$ for $i \neq j$, the set of relevant documents D_{REL}, the binary function $rel : D \rightarrow \{0, 1\}$ mapping relevant documents to 1 and non-relevant to 0 and P_k as precision at cutoff level k, AP is computed as follows:

$$AP(R) = \frac{\sum_{i=1}^{l} P_i \cdot rel(d_{r_i})}{|D_{\text{REL}}|}$$

Mean average precision (MAP) averages AP over all topics and can be used to evaluate the overall performance of an IR system.

A common feature of measures such as MAP — others are bpref [Buckley and Voorhees, 2004] and infAP [Yilmaz and Aslam, 2006] — is that they are primarily focused on measuring retrieval performance over the entire set of retrieved documents for each query, up to a pre-determined maximum (usually 1000). As already mentioned, such evaluation measures are a reasonable choice for scenarios in which users require as many relevant documents as possible. However, it is likely that users will not read all 1000 retrieved documents provided by a given IR system. For this

reason, other measures have been proposed to assess the correctness of the retrieval system given the fact that users typically only examine a limited set of (top-ranked) documents. For example, precision can be calculated at a given rank (denoted P@r). Precision at cut-off rank 10 (P@10) is commonly used to measure the accuracy of the top-retrieved documents.

When it is important to get the single top-ranked document correct, mean reciprocal rank (MRR) is another established evaluation measures. MRR is defined by the inverse rank of the first retrieved relevant document, averaged over all topics.

In some cases, relevance assessments contain multiple levels of relevance. Measures such as normalized discounting cumulative gain (NDCG) [Järvelin and Kekäläinen, 2000] can then be applied as they take into account the preference to have highly-relevant documents ranked above less relevant ones.

II.5.4 Established Datasets

IR experiments become reproducible and results comparable by re-using shared datasets consisting of a common corpus of documents, topics / queries and relevance assessments. In the field of IR, different evaluation initiatives defining various retrieval tasks and providing appropriate datasets have emerged.

Apart from datasets published by evaluation campaigns, parallel corpora are also of high interest to CLIR and MLIR. They are used as language resources, for example in order to train SMT systems or in order to identify cross-language latent concepts as in LSI. Additionally, they are also used as test collections, for example in mate retrieval scenarios.

Evaluation Campaigns:

Text REtrieval Conference (TREC) is organized yearly with the goal of providing a forum where IR systems can be systematically evaluated and compared. TREC is organized around different tracks (representing different IR tasks such as ad hoc search, entity search or search in special domains). For each track, datasets and topics / queries (and relevance judgments) are typically provided that participants can use to develop and tune their systems. Since its inception in 1992, TREC has been applying the pooling technique that allows for a cross-comparison of IR systems using incomplete assessments for test collections. TREC and similar conferences are organized in a competitive spirit in the sense that different groups can compete with their systems on a shared task and dataset, thus making results comparable. Such shared evaluations have indeed contributed substantially to scientific progress in terms of understanding which retrieval models, weighting methods etc. work better compared to others on a certain task. However, the main goal of TREC is not only to foster competition, but also to provide shared datasets to the community as a basis for systematic, comparable and reproducible results. The main

focus of TREC is monolingual retrieval of English documents, such that the published datasets only consist of English topics and documents.

Cross-lingual Evaluation Forum (CLEF) was established as the European counterpart of TREC with a strong focus on multilingual retrieval. In the ad hoc retrieval track, different datasets have been used between 2000 and 2009, such as a large collection of European newspapers and news agency documents with documents in 14 languages, the TEL dataset containing bibliographic entries of the European Library in English, French and German and a Persian newspaper corpus. For all datasets, topics in different languages are available, which makes these datasets suitable for CLIR and MLIR. The TEL dataset also contains mixed documents with fields in different languages.

NII Test Collection for IR Systems (NTCIR) defines a series of evaluation workshops that organize retrieval campaigns for Asian languages including Japanese, Chinese and Korean. A dataset of scientific abstracts in Japanese and English as well as news articles in Chinese, Korean, Japanese and English with topics in different languages has been released. In addition, a dataset for Japanese-English patent retrieval has been published.

Forum for Information Retrieval Evaluation (FIRE) is dedicated to Indian languages. It has released corpora built from web discussion forums and mailing lists in Bengali, English, Hindi and Marathi. Topics are provided in Bengali, English, Hindi, Marathi, Tamil, Telugu and Gujarati.

Workshop on Cross-lingual Expert Search (CriES) was held at the CLEF conference 2010. As part of the workshop, a pilot challenge on multilingual Expert Search was defined. The dataset was based on an official crawl of the Yahoo! Answers site, consisting of questions and answers posted by users of this portal. Topics in English, German, French and Spanish were defined and manual relevance assessments were published based on a result pool of submitted runs.

Parallel corpora:

JRC-Acquis is a document collection extracted from the Acquis Communautaire, the total body of European Union law that is applicable in all EU member states. It consists of parallel texts in the following 22 languages: Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Maltese, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene and Swedish.
`http://langtech.jrc.it/JRC-Acquis.html` (last accessed April 8, 2011)

Multext Dataset is a document collection derived from the Official Journal of European Community in the following five languages: English, German, Italian

Spanish and French.
`http://aune.lpl.univ-aix.fr/projects/multext/` (last accessed April 8, 2011)

Canadian Hansards consists of pairs of aligned text chunks (sentences or smaller fragments) from the official records (Hansards) of the 36th Canadian Parliament in English and French.
`http://www.isi.edu/natural-language/download/hansard/` (last accessed April 8, 2011)

Europarl is a parallel corpus containing the proceedings of the European Parliament from 1996 to 2009 in the following languages: Danish, German, Greek, English, Spanish, Finnish, French, Italian, Dutch, Portuguese and Swedish.
`http://www.statmt.org/europarl/` (last accessed April 8, 2011)

Wikipedia as a Corpus. Snapshots of the Wikipedia databases can be used as language resources. Actually, we showed that the language models derived from the English Wikipedia are similar to the language models derived from other established English datasets. This allows the conclusion that the Wikipedia databases in other language could also be used as resources for language models [Vrandecic et al., 2011]. Currently, Wikipedia supports approx. 260 languages.
`http://dumps.wikimedia.org/` (last accessed April 8, 2011)

II.5.5 References to this Thesis

The experiments presented in this thesis will be based on several of the introduced datasets. The model presented in Chapter IV will be evaluated using parallel datasets, namely JRC-Acquis and Multext. Further we participated in the ad hoc challenge defined by CLEF using this retrieval approach.

The creation of the dataset used in the CriES pilot challenge will be described in Chapter V. This includes the preprocessing of the raw data provided by Yahoo, the selection of topics and the creation of the gold standard based on result pooling and manual relevance assessments. Our proposed retrieval model in Chapter V will also be evaluated using the CriES dataset.

II.6 Tools, Software and Resources

The development of a complete IR system includes many different aspects, such as the implementation of preprocessing steps, file structures for inverted indexes and efficient retrieval algorithms. Building a system from scratch therefore constitutes an enormous effort. It is essential to build on existing tools to reduce the costs related to implementation.

In a specific project, it might be the case that only the retrieval model or ranking function need to be adapted, while the other components of the system can be used

off-the-shelf. Fortunately, there are different libraries providing standard IR components or even complete frameworks where certain components can be replaced.

In the following, we present selected tools and software libraries supporting the development of IR systems. We focus on established tools that are widely used and also have community support. The most popular IR framework is Lucene, which also contains wrappers for many other tools we present.

Preprocessing.

Content Analysis Toolkit (Tika): Toolkit to extract text from documents of various file types, e.g. PDF or DOC, implemented in Java. The detection of file types is also supported. Tika evolved from the Lucene project.
`http://tika.apache.org/` (last accessed April 8, 2011)

Snowball: Stemmer for several European languages. The implementation is very fast and also supports stop word removal. Lists of stop words for the supported languages are provided on the project web site.
`http://snowball.tartarus.org/` (last accessed April 8, 2011)

HTML Parser: Tool for parsing HTML documents. This can be used to extract textual content from Web sites, ignoring tags and parts not related to the semantic content.
`http://htmlparser.sourceforge.net/` (last accessed April 8, 2011)

BananaSplit: Compound splitter for German based on dictionary resources.
`http://niels.drni.de/s9y/pages/bananasplit.html` (last accessed April 8, 2011)

Translation. The web portal *Statistical Machine Translation*[4] is an excellent entry point to get information about Statistical Machine Translation systems. It provides software and datasets to train translation models.

As an example of a commercial SMT system, the Google Translate Service[5] provides an API for translation into various languages. However, as translation is part of preprocessing and is usually not deeply integrated into the retrieval framework, any commercial translation system might be plugged into a CLIR or MLIR system.

IR Frameworks.

Lucene is a widely used IR framework implemented in Java. It is available as Open Source software under the Apache License and can therefore be used in both commercial and Open Source programs. It has reached a mature development

[4] `http://www.statmt.org` (last accessed April 8, 2011)
[5] `http://translate.google.com/` (last accessed April 8, 2011)

status and is used in various applications. The main features of Lucene are scalability and reliability. This comes at the price of a decreased flexibility making it more difficult to exchange components. For instance, in Lucene the index construction is dependent on the retrieval model selected, so that the retrieval model can not be exchanged without rebuilding the index.
`http://lucene.apache.org/` (last accessed April 8, 2011)

Terrier and Lemur are tools used for research purposes. Terrier (implemented in Java) and Lemur (implemented in C++) are both flexible IR frameworks that can be easily extended and modified. Due to this different focus they do not match the stability and performance of Lucene.
`http://terrier.org/` (last accessed April 8, 2011)
`http://www.lemurproject.org/` (last accessed April 8, 2011)

Evaluation.

trec_eval is a tool that can be used to compute various evaluation measures for a given document ranking with respect to a gold standard. The input is expected as plain text files with simple syntax. Creating output in the TREC format enables to use trec_eval for any IR system. The IR frameworks presented above also support output in the TREC format.
`http://trec.nist.gov/trec_eval/` (last accessed April 8, 2011)

Chapter III

Semantics in Information Retrieval

In this chapter, we will present an overview of how semantics can be exploited in IR systems. Firstly, we will classify the different approaches presented in related work according to the used document representation model. As many of these document representations depend on *semantic relatedness*, we will then present the most common approaches to define semantic relatedness based on background knowledge. Finally, we will present semantic retrieval models that directly make use of semantics in the relevance function.

The semantic retrieval models presented in this chapter describe the foundations of the proposed retrieval models presented in Chapter IV and V. The overview of alternative approaches to exploit semantics in IR that is given in this chapter will also support the comparison and differentiation of our proposed models to related work.

III.1 Semantic Vector Spaces

In Chapter II, we defined the Vector Space Model (VSM) for representing documents. We also showed how vector similarities are used as retrieval models for IR. In literature, different approaches are known that extend the VSM using background knowledge. These approaches define *semantic vector spaces* that are spanned by topics or concepts. A *mapping function* from the term vector space to the semantic vector space allows to map term vectors of documents or queries to concept vectors. Applied to IR, the actual retrieval step in semantic vector spaces is similar to retrieval in the standard VSM. In both cases, the ranking of documents are based on vector similarity measures that are used to compare query vectors to document vectors.

In the following, we present the most prominent approaches to semantic vector spaces. They differ in the definition of concepts and in the type of background

knowledge that is used to extract these concepts. A main criteria for classifying these approaches is their usage of *intrinsic* or *external* background knowledge. Intrinsic knowledge is mined from the dataset the model is applied to — for example statistics about co-occurrences of terms in the corpus. External knowledge is provided by external sources that are in most cases not part of the current retrieval scenario.

III.1.1 Generalized Vector Space Model

The Generalized Vector Space Model (GVSM) was defined by Wong et al. [1985] to overcome some of the drawbacks of the standard VSM. The standard VSM is based on the *independence assumption* of terms. Occurrences of terms in documents are independent, which is modeled by the orthogonality of the vectors of all terms in the vocabulary. Formally, the scalar product of the term vectors $\vec{t_i}, \vec{t_j}$ of any two non-equal terms t_i, t_j in vocabulary V is zero:

$$\langle \vec{t_i}, \vec{t_j} \rangle = 0 \quad \text{for} \quad t_i, t_j \in V, i \neq j$$

Documents $d_k \in D$ are represented by term vectors with statistical weights for each term, for example the term frequency in the document. This leads to the following vector representation of document d:

$$\vec{d} = \sum_{t_i \in V} a_{d,i} \vec{t_i}$$

with $a_{d,i}$ being the weight of term t_i in document d.

For IR, the query vector $\vec{q}$ of query q is defined in the same way:

$$\vec{q} = \sum_{t_i \in V} a_{q,i} \vec{t_i}$$

with $a_{q,i}$ being the weight of term t_i in query q.

In the retrieval scenario, document and query vectors are compared using vector similarity measures. A standard measure is the cosine similarity which is the normalized scalar product in the vector space. Due to the independence of term vectors, this scalar product can be simplified, resulting in the following scoring function:

$$\text{score}(q, d) = \frac{\langle \vec{q}, \vec{d} \rangle}{\|\vec{q}\| \|\vec{d}\|} = \frac{\sum_{t_i \in V} a_{q,i} a_{d,i}}{\sqrt{\sum_{t_i \in V} a_{q,i}^2} \sqrt{\sum_{t_i \in V} a_{d,i}^2}}$$

The problems arising with the independence assumption are demonstrated by the following example:

Example III.1 *We consider the following example document d_1 and example query q_1:*

```
d1: Safety instructions for driving automobiles in
    winter conditions.
q1: car snow
```

Matching query q_1 to document d_1 in the standard VSM results in zero similarity as there is no term overlap. The term vectors of `automobile` *and* `car` *are orthogonal which contrast their synonymy relation. The terms* `winter` *and* `snow` *are related which is again not modeled in the VSM.*

To address this problem Wong et al. [1985] suggested to use the correlation matrix G to model the correlation of terms:

$$G = \begin{bmatrix} \vec{t_1} \cdot \vec{t_1} & \vec{t_1} \cdot \vec{t_2} & \cdots & \vec{t_1} \cdot \vec{t_n} \\ \vec{t_2} \cdot \vec{t_1} & \vec{t_2} \cdot \vec{t_2} & \cdots & \vec{t_2} \cdot \vec{t_n} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{t_n} \cdot \vec{t_1} & \vec{t_n} \cdot \vec{t_2} & \cdots & \vec{t_n} \cdot \vec{t_n} \end{bmatrix}$$

Given this correlation matrix, the score of document d given query q is then defined as:

$$\text{score}(q, d) = \vec{q}^T G \vec{d}$$

Example III.2 *Using the terms from the document and query presented in Example III.1, a possible correlation matrix is given by:*

$$G = \left[\begin{array}{r|ccccc} & t_1 & t_2 & t_3 & t_4 & t_5 \\ \hline \textit{automobile} = t_1 & 1 & \mathbf{1} & 0 & 0 & 0 \\ \textit{car} = t_2 & \mathbf{1} & 1 & 0 & 0 & 0 \\ \textit{instruction} = t_3 & 0 & 0 & 1 & 0 & 0 \\ \textit{snow} = t_4 & 0 & 0 & 0 & 1 & \mathbf{.5} \\ \textit{winter} = t_5 & 0 & 0 & 0 & \mathbf{.5} & 1 \end{array}\right]$$

As the entries in the correlation matrix of the synonym terms automobile *and* car *are set to 1, document d_1 will be a positive match for query q_1. In addition, the semantic relatedness of* snow *and* winter *is modeled using a correlation value of* .5.

An open question remains how the correlation matrix G is constructed. A popular automatic approach is using co-occurrence statistics of terms in training corpora [Wong et al., 1985]. There are various alternative approaches known in literature, many of them modeling the correlation using measures of semantic relatedness which will be described in detail in Section III.2. Recently, Anderka and Stein [2009] showed that Explicit Semantic Analysis — which forms the foundation of our research presented in Chapter IV — can also be interpreted as an instance of the GVSM, modeling term correlation using textual descriptions of concepts.

III.1.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a well known approach for extracting concepts from a given text corpus [Deerwester et al., 1990]. The resulting document representation intends a two-fold improvement in respect to the standard VSM. Firstly, related terms are mapped to the same concepts, which solves the problems presented in Example III.1. Secondly, the dimension of the vector space is reduced which allows for more efficient indexing and retrieval techniques.

LSI is based on Singular Value Decomposition (SVD) — a technique from Linear Algebra. A full SVD is a loss-free decomposition of a matrix M, which is decomposed into two orthogonal matrices U and V (left and right singular vectors) and a diagonal matrix Δ (singular values):

$$M = U\Delta V^T$$

Estimating less singular values and their corresponding singular vectors leads to an approximation of M by: $M \approx \tilde{U}\tilde{\Delta}\tilde{V}^T$.

Applied to vector space representations of documents, the matrix M is defined as the term-document matrix given corpus $D = \{d_1, \ldots, d_n\}$ and vocabulary $V = \{t_1, \ldots, t_m\}$. Entry $m_{i,j}$ then corresponds to the weight of term t_i in document d_j. Applying SVD to the term-document matrix results in a set of concepts Γ that is implicitly given by the columns of U. U covers the term-concept space by holding a weight for the correlation of each term-concept pair. Analogously, the document-concept space V contains a weight for the document-concept correlation. Since the dimension of Δ corresponds to the number of concepts and singular values are derived in descending order, a reduced SVD results in the most relevant concepts.

Example III.3 *We will use the following four documents in our example:*

```
d1: Safety instructions for driving automobiles
    in winter conditions.
d2: Cars, snow and winter.
d3: Snow forecast in winter time.
d4: Instructions to wash your automobile.
```

These documents define the following term-document matrix M:

$$M = \left[\begin{array}{r|cccc} & d_1 & d_2 & d_3 & d_4 \\ \hline \textit{automobile} & 1 & 0 & 0 & 1 \\ \textit{car} & 0 & 1 & 0 & 0 \\ \textit{instruction} & 1 & 0 & 0 & 1 \\ \textit{snow} & 0 & 1 & 1 & 0 \\ \textit{winter} & 1 & 1 & 1 & 0 \end{array}\right]$$

Applying SVD on M results in the following matrixes U, Δ and V^T, with γ_i representing the implicit concepts:

$$U = \begin{array}{r|cccc} & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \hline \textit{automobile} & -0.39 & 0.56 & 0.12 & 0.15 \\ \textit{car} & -0.23 & -0.28 & 0.89 & -0.25 \\ \textit{instruction} & -0.39 & 0.56 & 0.12 & 0.15 \\ \textit{snow} & -0.43 & -0.49 & -0.048 & 0.76 \\ \textit{winter} & -0.68 & -0.24 & -0.41 & -0.56 \end{array}$$

$$\Delta = \begin{bmatrix} 2.40 & 0 & 0 & 0 \\ 0 & 1.89 & 0 & 0 \\ 0 & 0 & 0.70 & 0 \\ 0 & 0 & 0 & 0.45 \end{bmatrix}$$

$$V^T = \begin{array}{c|cccc} & d_1 & d_2 & d_3 & d_4 \\ \hline \gamma_1 & -0.61 & -0.56 & -0.46 & -0.33 \\ \gamma_2 & 0.46 & -0.53 & -0.38 & 0.59 \\ \gamma_3 & -0.25 & 0.62 & -0.66 & 0.34 \\ \gamma_4 & -0.59 & -0.11 & 0.45 & 0.66 \end{array}$$

By reducing the dimension of the model, LSI brings related terms together and forms concepts. In this new space, documents are no longer represented by terms but by concepts. New documents (for example from the retrieval document collection) or queries are represented in terms of concepts by folding them in into the LSI model. This is done by multiplying their term-vector with $\tilde{U}$. The document-concept mapping is thus defined by the following function:

$$\vec{d^*} = \tilde{\Delta}^{-1}\tilde{U}^T\vec{d}$$

By mapping term vectors of queries in the same way, retrieval can be performed using vector similarity measures in the concept space. The knowledge that is used to define the concept space using LSI is given by the term-document matrix of a text corpus. In a retrieval scenario, this could be either the test collection or any other collection with sufficient vocabulary overlap to the test collection.

Example III.4 *Selecting the two largest eigenvalues in Δ from Example III.3 results in $\tilde{\Delta}$ with all other eigenvalues set to zero:*

$$\tilde{\Delta} = \begin{bmatrix} 2.40 & 0 & 0 & 0 \\ 0 & 1.89 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This implicitly defines two concepts γ_1 and γ_2. The topic-document matrix M^* is

then given as:

$$M^* = \tilde{\Delta}^{-1}\tilde{U}^T M = \left[\begin{array}{c|cccc} & d_1 & d_2 & d_3 & d_4 \\ \hline \gamma_1 & -3.50 & -3.20 & -2.65 & -1.87 \\ \gamma_2 & 1.66 & -1.91 & -1.37 & 2.11 \end{array}\right]$$

III.1.3 Latent Dirichlet Allocation

In the family of probabilistic latent topic models, Latent Dirichlet Allocation (LDA) defines a generative model that allows to represent documents based on a vector space defined by latent topics in analogy to LSI.

The basic idea of this approach is to abstract from particular words and to represent documents by mixtures over a set Γ of latent concepts $\gamma_1, \ldots, \gamma_k$ (for example hidden document-specific themes of interest), whereby each concept is characterized by a fixed conditional distribution over words in the vocabulary.

LDA assumes that all terms (both observed and previously unseen) are generated by randomly chosen latent concepts. In contrast to the SVD used in LSI, LDA has a well founded probabilistic background and tends to result in more flexible model fitting [Blei et al., 2003]. It allows resources to belong to multiple latent concepts with different degrees of confidence and offers a natural way of assigning probabilistic feature vectors to previously unseen resources.

In line with Blei et al. [2003], the content of the particular document is generated by selecting a multinomial distribution over concepts given the Dirichlet prior. For each term, a concept is generated from the document-specific concept distribution, and then a keyword is generated from the discrete distribution for that concept as follows:

1. The number of words in the document is chosen: $n \sim Poisson(\xi)$
2. The keyword generating parameter is chosen: $\theta \sim Dir(\alpha)$
3. For each of the document terms $t_1, \ldots, t_n$:
   - The generative concept for t_i is chosen: $c_i \sim Multinomial(\theta)$.
   - The term t_i is generated using a multinomial probability with parameter β conditioned on c_i: $P(t_i|c_i, \beta)$

Applied to IR, the LDA approach can be instantiated as follows. In the first step, the background knowledge collection B, which is either the test collection D or a different collection covering similar topics, is used for fitting the corpus-level properties α and β which are estimated using the variational Expectation Maximization (EM) procedure [Blei et al., 2003]. In the process of corpus analysis, we also obtain the document-level variables θ, sampled once per training document. As a result we obtain the posterior distribution of the hidden concepts $\gamma_1, \ldots, \gamma_k$ given a document d:

$$P_d(\theta, \vec{\gamma}|\vec{t}, \alpha, \beta) = \frac{P_u(\theta, \vec{\gamma}, \vec{t}|\alpha, \beta)}{P_u(\vec{t}|\alpha, \beta)} \tag{III.1}$$

Given the estimated model parameters, the similarity between documents and queries can be computed. Each document and query are mapped to a concept-based representation by Φ. In the LDA approach, Φ estimates the distribution of the hidden concepts $\gamma_1, \ldots, \gamma_k$ (III.1) using the variational EM procedure [Blei et al., 2003]. The estimated probabilities are considered as characteristic features of the document in the latent concept-based feature space: $\Phi(d) = P_d(\theta, \vec{\gamma}|\vec{t}, \alpha, \beta)$.

Example III.5 *As example for LDA we apply a collapsed Gibbs samplers initialized with two topics γ_1, γ_2 on the documents defined in Example III.3. This results in the following term-topic matrix:*

	γ_1	γ_2
instruction	0	2
automobile	0	2
winter	2	1
car	1	0
snow	2	0

The assignment of documents to topics is then defined as:

	d_1	d_2	d_3	d_4
γ_1	0	3	2	0
γ_2	3	0	0	2

Our expectation is that queries and their relevant documents show the similar behavior in terms of probabilities over latent concepts (see Equation III.1). We notice that the cosine similarity measure enforces in this case the intuitively expected IR-like similarity estimation behavior, whereby the similarity of the document to itself is maximized and the orthogonal document feature vectors (for example document vectors with disjoint sets of non-zero topic probabilities) have zero similarity.

Various extensions of LDA have been proposed in related work that allow to exploit background knowledge. For example Zhou et al. [2008] propose to build topic models based on social annotations that are then used in a retrieval framework. Other examples are presented in [Chemudugunta et al., 2008] or [Andrzejewski et al., 2009].

III.1.4 Semantic Smoothing Kernels

Bloehdorn et al. [2006] presented an approach that exploits background knowledge given by a taxonomy to define *Semantic Smoothing Kernels*. These kernels have been introduced by Siolas and d'Alché-Buc [2000] and are defined as follows:

Definition III.1 (Semantic Smoothing Kernel) *The semantic smoothing kernel for two data items (for example documents) x, y given by their vector representation $\vec{x}, \vec{y} \in X$ is given by*

$$\kappa(\vec{x}, \vec{y}) = \vec{x} Q \vec{y}^T$$

where Q is a square symmetric matrix whose entries represent the semantic proximity between the dimensions of the input space X [Bloehdorn et al., 2006].

Applied to text mining, Semantic Smoothing Kernels can be interpreted as instance of the GVSM with Q being the correlation matrix of terms. Bloehdorn et al. [2006] propose to use super-concept expansion based on WordNet as knowledge resource to define entries in matrix Q. We will present and analyze this and other established approaches to define semantic relatedness in detail in the following Section III.2.

Apart from the IR scenario described in this thesis, kernel functions can be applied to other text mining tasks such as classification or clustering. They offer therefore flexible tools to use semantic background knowledge in several application scenarios.

III.1.5 Explicit Semantic Analysis

In contrast to LSI or LDA, the semantic document representation defined by Explicit Semantic Analysis (ESA) is based on explicit concept spaces. Gabrilovich and Markovitch [2007] introduced ESA as a mapping of documents to a vector space that is spanned by a set of concepts. Each concept has a textual description, which is used to define the mapping function. In summary, the text similarity between a document and a concept description is used to compute the value of the dimension in the vector representation that corresponds to the according concept. The model of ESA and its application to IR will be presented in detail in Chapter IV. In this chapter, we will also present several examples of ESA.

Various knowledge sources for explicit concept definitions have been used in literature. The most prominent example is Wikipedia [Gabrilovich and Markovitch, 2007]. Concepts are thereby defined by Wikipedia articles. Another example is data from the Open Directory Project[1]. In this case, concepts correspond to Web sites or bundles of Web sites with similar topics [Gabrilovich and Markovitch, 2007]. Finally, Wiktionary[2] has also been used as background knowledge for ESA [Zesch et al., 2008].

Anderka and Stein [2009] have shown that ESA applied to IR can be interpreted as instance of the GVSM. The correlation matrix G as introduced in Section III.1.1 is thereby defined by term correlations that are exploited from the textual concept descriptions.

III.1.6 Ontology-based Document Representations

An approach for ontology-based document representations is presented by Baziz et al. [2005]. They propose to build *semantic networks* for documents. The network is constructed by matching terms to concepts that are defined in an ontology

[1] `http://www.dmoz.org/` (last accessed April 8, 2011)
[2] `http://www.wiktionary.org/` (last accessed April 8, 2011)

which is used as background knowledge. Each concept is thereby disambiguated and linked to other concepts using semantic relatedness measures (see the following Section III.2). For other approaches that match documents to ontologies for a semantic representation of documents see for example [Popov et al., 2004] or [Castells et al., 2007].

Applied to IR, the semantic networks of documents are used to map documents to the concept space. This concept space is spanned by concepts — similar to ESA. Weights of concepts for a document are defined by the frequency of concepts and by the weights of their relations in the semantic network.

III.2 Semantic Relatedness

Many IR systems that exploit semantics in their retrieval model are based on *semantic relatedness*:

Definition III.2 (Semantic Relatedness) *Semantic relatedness measures the lexical relatedness of concepts. Semantic relatedness extends the notion of similarity as it includes any kind of functional relationship [Budanitsky and Hirst, 2006]. Examples are "meronymy (*car-wheel*), antonymy (*hot-cold*) or frequent association (*pencil-paper, penguin-Antarctica, rain-flood*)" [Budanitsky and Hirst, 2006, pg. 13].*

Semantic relatedness is used in many fields of Natural Language Processing (NLP), including "word sense disambiguation, determining the structure of texts, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection, and the automatic correction of word errors in text" [Budanitsky and Hirst, 2006, pg. 13]. In this thesis we will focus on its application to IR. The following approaches have been proposed to extend IR models using semantic relatedness:

Term Correlation Matrix: As already mentioned in Section III.1, semantic relatedness has been used to construct the correlation matrix G as defined by the GVSM. Each entry g_{ij} of G — defined as the correlation $\vec{t_i} \cdot \vec{t_j}$ of term t_i and t_j — then directly corresponds to the semantic relatedness of t_i and t_j:

$$g_{ij} := \text{sr}(t_i, t_j)$$

This approach is for example used by [Gurevych et al., 2007] or [Müller et al., 2007]. Bloehdorn et al. [2006] defined the correlation matrix used to define semantic smoothing kernels in a similar way.

Query Expansion: As introduced in Chapter II, query expansion is a technique to improve the recall in retrieval scenarios. By extending the set of query terms, more documents are matched to a query. This aims at solving problems that are

for example introduced by synonymy of query and document terms. Semantic relatedness can be used to select appropriate terms for query expansion. This is supported by the study of Müller and Gurevych [2009]. They show that indeed semantic relatedness can be used to overcome the vocabulary mismatch problems between query terms and document terms. A further example of a retrieval approach that uses query expansion based on semantic relatedness is given by [Fang, 2008].

Topic Space: Semantic relatedness as relation between terms can be used to define equivalence classes of related terms. These equivalence classes (or topics) can then be used to define a topic space that allows a more compact representation of term vectors. Using correlation values mined from a background collection, LSI and LDA are examples for the application of semantic relatedness to the definition of semantic document representations.

A known problem of defining semantic relatedness is contextual dependence of the relation, given for example by the ambiguity of terms. "Lexical semantic relatedness is sometimes constructed in context and cannot always be determined purely from an *a priori* lexical resource" [Budanitsky and Hirst, 2006, pg. 45]. In retrieval scenarios it is therefore important to not only consider the pairwise relation of query and document terms, but to also include the context given by other query terms or the whole document.

Semantic similarity can be seen as a kind of semantic relatedness. While relations such as hyponomy/hypernomy will still be useful to define semantic similarity, other type of relations such as co-occurrences in text corpora will not.

III.2.1 Classification of Semantic Relatedness

Approaches to semantic relatedness can be classified according to the type of background knowledge used to define the relation:

Dictionary: Entries in dictionaries can be exploited to define semantic relatedness by using the provided references to other terms. Alternatively, the description of two terms can be compared and the relatedness is then defined by a text similarity measure such as term overlap.

Thesaurus: Similar to dictionaries, thesauri contain information about terms of a language. As they group together words of similar meanings they are a more practical resource for semantic relatedness.

Semantic Network: In semantic networks, terms are connected based on different kind of relations, for example hyponymy/hypernymy. The most prominent semantic network for English is WordNet. Various approaches have been presented to define semantic relatedness based on semantic networks which will be presented in Section III.2.2. Additionally, taxonomies and ontologies can be interpreted as semantic networks and will also be considered in this section.

Unstructured Text. In unstructured text, co-occurrences of terms are often interpreted as semantic relatedness. While this is clearly only an estimation of relatedness, the advantage lies in the large amount of available training data. Different definition of context can be used to mine co-occurrences, ranging from sentences over paragraphs to whole documents.

Lin [1998] proposed an alternative classification of semantic relatedness by analyzing the properties of the semantic relatedness function in a mathematical sense. He presents a set of criteria to evaluate these functions that are derived from the mathematical definition of metrics as well as from intuitions for semantic relatedness. In the following will use the classification based on knowledge sources. The mathematical properties of the semantic relatedness functions have less relevance to the topic of this thesis than the selection of appropriate knowledge sources.

III.2.2 Structured Knowledge Sources

Lexical structured knowledge sources define relations between concepts that are associated with terms. Strube and Ponzetto [2006] proposed the following definition of semantic relatedness on structured knowledge sources (or taxnonomies):

Definition III.3 (Semantic Relatedness in Taxonomies) *"Semantic relatedness indicates how much two concepts are related in a taxonomy by using all relations between them (i.e.* hyponymic/hypernymic*,* meronymic *and any kind of functional relations including* has-part, is-made-of, is-an-attribute-of, etc.*)" [Strube and Ponzetto, 2006, pg. 1419].*

Different knowledge sources have been exploited to define semantic relatedness. In the following we will present the most prominent and therefore widely used knowledge sources.

WordNet. WordNet is the most often used resource for the definition of semantic relatedness. Budanitsky and Hirst [2006] evaluated several definitions of semantic relatedness based on WordNet. This evaluation was based on a test set of word pairs that were manually judged for semantic relatedness by human assessors.

An alternative approach to evaluate semantic relatedness measures is to actually evaluate certain tasks that apply these measures. IR is an example for such a task. In this case, the relatedness measures are evaluated according to the improvements achieved in the retrieval task. It is important to mention that both evaluation approaches might lead to different rankings of relatedness measures. The human perception of relatedness will not be equivalent to the relatedness that might be useful for a specific retrieval task.

The following are examples for WordNet based measures of semantic relatedness:

- Resnik [1999] proposed a semantic relatedness measure based on subsuming elements. Given two concepts c_1, c_2 and a set $S(c_1, c_2)$ of subsuming concepts of c_1 and c_2, the relatedness between c_1 and c_2 is defined as follows:

$$\text{sr}(c_1, c_2) := \max_{c \in S(c_1, c_2)} [-\log P(c)]$$

  whereas $P(c)$ is the probability of concept c, that is defined as 1 for the root concepts and decreasing when moving down the taxonomy.

- Lin [1998] propose a definition of semantic relatedness based in information theory. He models his relatedness measure as the ratio of the amount of information that is needed to state the commonality of two concepts c_1 and c_2 to the amount of information that is needed to fully describe both concepts.

- Jiang and Conrath [1997] combine the information theoretic measure of semantic similarity presented by Resnik [1995] with corpus statistical information. In particular, they define weights for edges in semantic networks based on local structures and define the similarity by the aggregated weight of the shortest path between two concepts.

Wikipedia. Wikipedia — interpreted as structured knowledge source — has also be used to define semantic relatedness measures. The structure is thereby given by links that connect articles (*page links*), links that define the category structure (*category links*) and links that connect articles or categories across languages (*cross-language links*). The following are examples for semantic similarity measures defined on the structure of Wikipedia:

- Strube and Ponzetto [2006] propose to match terms to titles of Wikipedia pages. Using the category structure of Wikipedia, the relatedness measure is then computed based on the category paths found between matching articles.

- Witten and Milne [2008] define semantic relatedness using the link structure established by links between articles. In contrast to Strube and Ponzetto [2006], they do not use information about categories.

III.2.3 Text Corpora as Knowledge Source

Apart from using structured knowledge sources to define semantic relatedness, other approaches are based on unstructured resources. The biggest advantage of text based approaches is the large amount of available data. Semantic networks as presented above are expensive in creation and maintenance. For relatedness measures based on term co-occurrences, in principle any text corpus can be used as knowledge source.

For the majority of approaches, the most important information extracted from the training corpus are term correlations. Usually correlation values are mined using statistical measures on co-occurrences of terms. While structured knowledge sources

are often used to define semantic similarity, text based approaches based on term correlations naturally define semantic relatedness, as co-occurrences of terms are hints for relatedness and do not allow to infer similarity.

An example for a distributional measure of semantic similarity is given by Mohammad and Hirst [2006]. They also propose to cluster the documents used as background knowledge to categories, which allows for a more efficient implementation.

Gabrilovich and Markovitch [2007] have introduced Explicit Semantic Analysis (ESA) that is based on text corpora — such as corpora derived from Wikipedia, the Open Directory Project or Wiktionary — to compute semantic relatedness. In contrast to Strube and Ponzetto [2006] or Witten and Milne [2008], the structure is not primarily used to compute the relatedness. Terms are matched against the body of articles leading to vector representations of terms in a the concept space spanned by all articles. More details on this technique will be given in Chapter IV. In addition, Gabrilovich and Markovitch [2007] used some structural information in Wikipedia to compute *a priori* weights of articles, which is in general not possible for text corpora.

III.3 Semantic Retrieval Models

Defining semantic document representations is one possible approach to use background knowledge in IR systems. As described above, the main problems thereby are the definition of the set of concepts and the definition of the mapping function that allows to map term vectors of documents and queries to the concept space.

Alternatively, semantic knowledge can directly be integrated in the retrieval model. In the following, we will describe three approaches of semantic retrieval models:

III.3.1 Language Models

Language Models define retrieval models based on generative models of documents (see Chapter II). These models are described by probability distributions of terms t_i being generated by a document d: $P(t_i|d)$. *Mixture models* of probability allow the integration of different sources of evidence to model the probability of term t_i given document d:

$$P^*(t_i|d) = \alpha_1 P_1(t_i|d) + \alpha_2 P_1(t_i|d) + \cdots + \alpha_k P_k(t_i|d)$$

with $\sum_i \alpha_i = 1$. Each conditional probability $P_i(t_i|d)$ can be modeled using different knowledge. Examples are the relative term frequency of term t_i in document d or the *a priori* probability of t_i in the corpus. By using other background knowledge to model P_i, semantics can be integrated into IR based on language models.

An example for such modeling is presented by Cao et al. [2009]. They propose to use a mixture model that exploits categorization information to improve retrieval on

a dataset from a Social Question/Answer Site. The semantic knowledge is given by the categories of each questions, which is then used to define the mixture language model. These experiments are related to our research presented in Chapter V.

III.3.2 Learning to Rank

Supervised Machine Learning (ML) is defined as learning models based on training data that allow to make predictions about new data. For example in the case of classification, the training data consists of instances that are assigned to a set of classes. The learned model based on this training data then predicts the class assignment of new instances.

Different approaches exist to use ML methods for IR, which are often labeled as *learning to rank* approaches. They differ in the training data that is used to learn the model and in their integration in the retrieval model. In line with the main topic of this thesis, the training data can be referred to as background knowledge. Ranking models that are optimized using ML techniques allow to exploit this knowledge in the training process and can therefore be classified as semantic retrieval models.

Given a feature representation of documents and a query q, classifiers can directly be applied as retrieval function. Given a binary relevance model, there are two target classes: *relevant* and *non-relevant*. The scoring function is then defined as:

$$\text{score}(d, q) = P(F_q(d) \rightarrow \text{relevant})$$

with $F_q(d)$ being the feature representation of document d given query q. Training data (or background knowledge) is given by relevance assessments of training queries. This approach to integrate ML in IR will be used in the discriminative model we present in Chapter V.

An alternative approach is presented by Joachims [2002]. He proposes to classify pairs of documents given a query. The classifier then predicts the relative ranking of these two documents. These predictions can then be used to define a final ranking of all documents. As training data, he uses click through data collected by an Internet search engine. This data motivates the classification of pairs of documents as this reflects the behavior of users. Given the list of top ranked documents, users click on the most relevant ones and therefore prefer them to the other presented documents. More details on this approach will be presented in Section V.4.3.

III.3.3 Extended Query Models

In addition to defining semantic representation of documents, *extended query models* have been proposed as another approach to use background knowledge for IR. Extended query models can be classified into manual and automatic approaches. The goal is to define a query representation that provides a precise model of the user's information need.

Manual Approaches. Manual approaches depend on the user to provide information that allows to build more detailed query models. For example in the retrieval system presented by Rinaldi [2009], the user is asked to provide subject keywords and domain keywords. Subject keywords are matched to the text content of the documents in the test collection. Domain keywords are used to build a *dynamic semantic network* using background knowledge provided by WordNet. This network is then used to refine search results. Other manual approaches to build extended query models are presented in [Chen and Dhar, 1990] or [Anick, 2003].

Automatic Approaches. Automatic approaches to build extended query models include Pseudo Relevance Feedback (PRF) as presented in Chapter II. Using a first retrieval step, the top documents are assumed to be relevant and additional keywords are extracted from these documents. In a second retrieval step, the extended query is matched against the test corpus.

While PRF is based on unstructured text — often the test collection itself — other approaches to automatic query expansion are based on structured background knowledge. Building on the notation of semantic relatedness as introduced above, related terms in respect to query terms can be used to expand the query. Any definition of semantic relatedness could thereby be used to select the expansion terms, exploiting different sources of background knowledge such as thesauri or semantic networks.

Chapter IV

Cross-lingual Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) has been proposed in recent years as a successful approach to concept indexing by authors such as Gabrilovich and Markovitch [2007], Gupta and Ratinov [2008], Müller and Gurevych [2008] or Potthast et al. [2008]. Based on a given set of concepts with textual descriptions, ESA defines the representation of documents in respect to these concepts. This can be applied for example to compute Semantic Relatedness measures or in retrieval tasks.

The strength of the ESA approach is founded in its definition of concepts. ESA does not depend on resources that are traditionally used for concept indexing, for instance semantic term networks such as WordNet, taxonomies or ontologies. In contrast, ESA allows to exploit unstructured or semi-structured text corpora to define the set of concepts that is then used for concept indexing.

This definition of concepts facilitates to use the large amount of already available text corpora as background knowledge which includes datasets of Web 2.0 documents. Many of these datasets contain documents in multiple languages, cover a broad range of topic fields and are constantly updated. These are clear advantages compared to the usage of traditional resources which are usually hand-crafted and therefore lack these features.

Web 2.0 datasets have the problem that they contain many noisy documents. These are for instance spam documents or documents that do not describe a specific topic but contain recordings of the communication between users. In Wikipedia, an example for such documents are the discussion pages that are attached to each article. Further, Web 2.0 datasets are not carefully designed and populated such as hand-crafted semantic resources but are built by the contributions of many heterogeneous Internet users. However, ESA is based on statistical measures and inconsistencies or even errors in the background knowledge have only small influence. This allows to exploit these datasets as the benefits of having multilingual, broad and up-to-date

resources overweight the drawback of the presence of some level of noise in the data.

As an outstanding Web 2.0 dataset, Wikipedia combines all the features mentioned above. It is published in various languages, has content about many topics and is constantly updated. Additionally, the level of noise in Wikipedia is low compared to other Web 2.0 resources. Further, Wikipedia has features that introduce structural knowledge and can be used to design extensions of the ESA model. Examples of these features are the category structure of Wikipedia or links between Wikipedia articles. By reason of these characteristics of Wikipedia, most of the published approaches to ESA use Wikipedia as background knowledge — which is also pursued in this chapter. Some of our proposed extensions to ESA also exploit the information that is contained in the structure of Wikipedia.

In the following, we will present a generalized model of ESA. This models allows different design choices, which will be evaluated in our experiments. The most important extension is the definition of Cross-lingual Explicit Semantic Analysis (CL-ESA), which allows to apply ESA on text in multiple languages. We will use Information Retrieval (IR) as the application scenario of CL-ESA in our experiments. Alternative applications such as computing Semantic Relatedness measures will not be covered in this thesis.

IV.1 Preliminaries

Wikipedia is an interesting resource for concepts. These concepts are thereby defined by articles or categories and are related to each other by links in Wikipedia. For this reason, Wikipedia is often used as background knowledge of ESA that basically requires a set of concepts with textual descriptions.

In this section, we will define all features of Wikipedia that are important to understand how ESA can exploit Wikipedia as background knowledge. Further, we introduce a running example that is used to illustrate the different semantic document representations.

Despite of using Wikipedia, ESA is a general model and is able to exploit other resources as background knowledge. The extensions to ESA that we propose in this chapter depend on structures that connect concepts across languages. Some of them also make use of a hierarchical category structure on concepts. These features are given in Wikipedia by cross-language links and category links. However, any dataset containing multilingual textual descriptions of topics that can be regarded as concepts could potentially be used as background knowledge for CL-ESA.

Wikipedia Database. For our retrieval framework, we consider a fixed set of supported languages $L = \{l_1, \ldots, l_n\}$. As background knowledge for the multilingual retrieval model, we use Wikipedia databases W_l with $l \in L$. W_l consists of a set of articles $\mathcal{A}(W_l)$ and a set of categories $\mathcal{C}(W_l)$ in language l.

In this chapter, we will use the sample subset of English and German articles and categories that is presented in Figure IV.1 to give examples for the different ESA

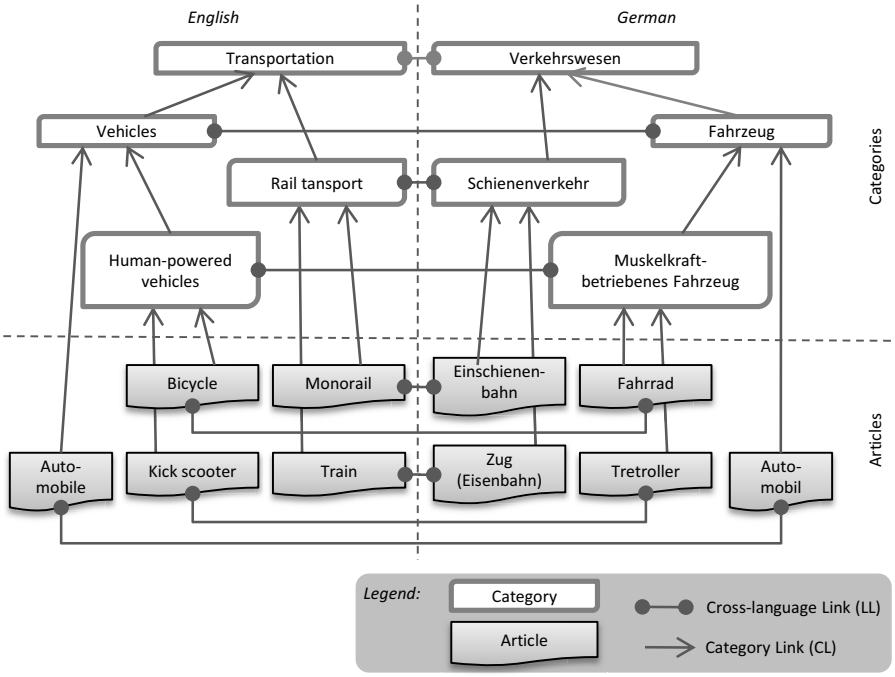


Figure IV.1: Wikipedia articles, categories and link structure that are exploited for document representations in different concept spaces (ESA, CL-ESA, Cat-ESA and Tree-Esa).

approaches. Articles are presented in the lower level and categories in the upper level. In Wikipedia, there are links that assign articles to categories as well as links between categories defining sub-category relationships. We will refer to both as category links (CL), represented by vertical arrows in Figure IV.1. Cross-language links (LL), represented in Figure IV.1 by horizontal connections, connect equivalent articles or categories across languages and are therefore crucial for a cross-lingual extension of ESA.

ESA is based on a set of concepts as background knowledge. In this chapter, we propose different approaches to exploit Wikipedia to define concepts. The most simple model is only based on articles in one language. The multilingual extension of this model also uses articles in other languages and exploits cross-language links between articles. Other models are based on categories and subtrees of categories. When describing these models in Section IV.2 and IV.3, we will again refer to Figure IV.1 that visualizes the structure of Wikipedia used to define each of the different concept spaces.

The Wikipedia databases contain user-generated content that is collaboratively edited. All links, category and cross-language links in particular, are manually created and therefore not always consistent across languages. We identified two different types of such inconsistencies that are most relevant in our context.

First, there is the problem of missing or false language links. An example for such a missing link is the article `Container ship`:

Example IV.1 (Inconsistent LLs) *In the English Wikipedia, the article* `Container ship` *is not linked to its Spanish equivalent* `Buque portacontenedores`.

Second, categories in one language potentially contain articles that are not part of the equivalent category in another language. A similar problem are differences in sub-category relationships. An example is given by the category `Natural scientists`:

Example IV.2 (Structural Differences of Categories across Languages) *In the English Wikipedia, the category* `Natural scientists` *is a sub-category of* `Natural sciences`. *In the German Wikipedia however, the equivalent articles* `Naturwissenschaftler` *and* `Naturwissenschaft` *have no such sub-category relation.*[1]

While manually created links might be erroneous, it has been shown that the high number of Wikipedia editors as a collective succeeds very well in maintaining information quality at high standards [Giles, 2005]. In Chapter VI, this behavior will be analyzed in respect to missing cross-language links. As ESA relies on statistical term

[1] This structural error is present in the Wikipedia dump of September 2009 that was used in our experiments. In the current online version of Wikipedia, the missing category link was added. This is an example of how Wikipedia is constantly improved and errors are detected and solved by the community.

distributions across a high number of articles or categories, we assume that these *errors* — actually structural differences across languages — will only marginally affect the retrieval performance. In Chapter VI, we will also show that the Wikipedia community is able to identify errors in the cross-language link structure and that Wikipedia is therefore constantly improving over time.

Running Example. The different aspects and different variants of ESA and the application of ESA to retrieval will be illustrated using several examples. These example are all based on a sample query and a set of sample concepts derived from Wikipedia articles and categories. These concepts and their relations are presented in Figure IV.1. The visualization of the structural features that are presented in this figure will help to understand the extensions of ESA that we define in this chapter.

In addition to the sample set of concepts, we will use the following sample query for the examples in the next sections:

Example IV.3 (Sample Query for Running Example) *The title of a document from the Multext dataset*[2] *(introduced in Section IV.5) is used as running example. We use the translation into English and German:*

```
English: The transport of bicycles on trains
 German: Beförderung von Fahrrädern mit dem Zug
```

IV.2 Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) indexes a given document d with respect to a set of explicitly given external concepts. Gabrilovich and Markovitch [2007] have outlined the general theory behind ESA and in particular described its instantiation to the case of using Wikipedia articles as external concepts. In this thesis, we will basically build on this instantiation of ESA, which we describe in more detail in the following.

IV.2.1 Definition of Concepts

All concept models described below build on an explicitly defined and finite set of concepts $C = \{c_1, \ldots, c_m\}$ with respect to which documents will be indexed.

Further, for all models we assume the existence of a text signature τ of concepts which defines the textual description of concepts. As we extract concepts from Wikipedia, this is a (sub-)set of all articles $\mathcal{A}(W)$ of Wikipedia database W:

$$\tau : C \rightarrow 2^{\mathcal{A}(W)}$$

[2] Document ID: `FXAC93006DEC.0012.02.00`

By considering the textual signature of concepts, we are able to compute the term distribution for each concept. Using term weighting techniques, this information is then used to index documents with respect to concepts.

IV.2.2 Original ESA Model

ESA maps a document d to a high-dimensional real-valued vector space spanned by concepts C. This mapping function $\Phi : D \rightarrow \mathbb{R}^m$ is defined as follows:

$$\Phi(d) := (\phi(d, c_1), \ldots, \phi(d, c_m))^T \tag{IV.1}$$

The value $\phi(d, c_i)$ in the ESA vector of d expresses the strength of association between a document d and concept c_i. This value is computed based on term distributions in d and in the text signature $\tau(c_i)$ of concept c_i. In the original model, it is based on a tf.idf function on the text signature of concepts applied to all terms t of document d:

$$\phi(d, c) := \sum_{t \in d} \text{tf.idf}_{\tau(c)}(t)$$

The tf.idf function is based on the Bag-of-Words model, where each dimension of the term vector corresponds to a unique term. For a given document d in corpus D, the values of each dimension represent the weight of a term t in d. This weight is typically computed by taking into account the distribution of the term in the document and corpus. For instance, tf.idf is a widely used function counting the occurrences of a term in a document weighted by the inverse number of documents in the corpus which have at least one occurrence of this term:

$$\text{tf.idf}_d(t) = \frac{\text{tf}_d(t)}{|d|} \log \frac{|D|}{\text{df}(t)}$$

with $\text{tf}_d(t)$ as the term frequency of t in d, $|d|$ the number of tokens in d, $|D|$ the number of documents and $\text{df}(t)$ the document frequency of t, defined by the number of documents in D containing term t.

For our running example, the association strength to the concept `Bicycle` is visualized in Figure IV.2. In order to facilitate understanding, we use the simple term frequency values instead of the term frequencies weighted by the document frequencies (tf.idf). The term frequency vector is then based on the number of occurrences of each term, which is given by $\text{tf}_{\tau(\textit{Bicylce})}(t)$. In this case, the association strength is computed by summing all values in this frequency vector that correspond to terms occurring in the sample text.

IV.2.3 ESA applied to IR

ESA allows to map arbitrary text to the vector space defined by the set of concepts. As described in Chapter II, these vector representations can be used in a concept-based retrieval framework.

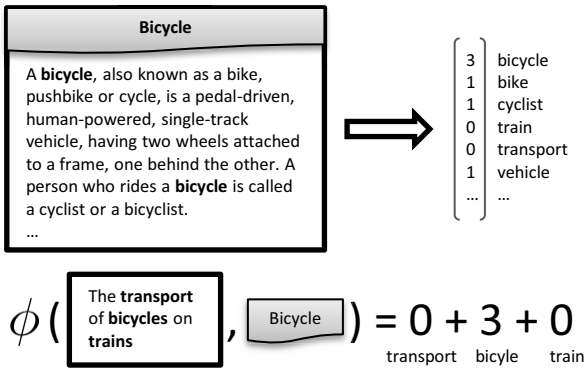


Figure IV.2: Association strength between the example text and the concept *Bicycle*, described by the according Wikipedia article.

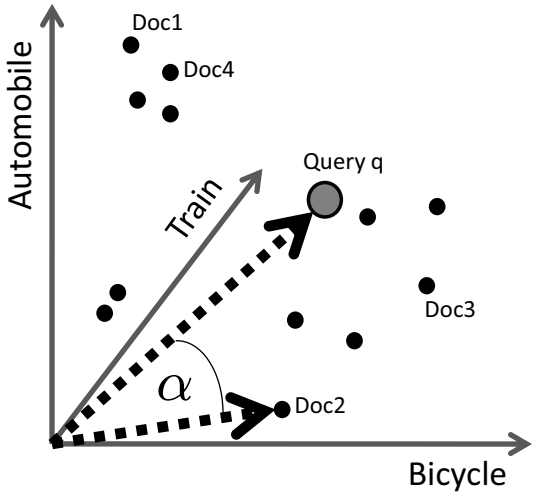


Figure IV.3: Retrieval in concept space. The similarity of the query to documents is defined by the angle between their concept vector representations, which is for example computed by the cosine function.

Given query q and document collection D, ESA-based IR is defined by the following steps:

1. The query q is mapped to the concept space, resulting in the vector representation:
$$\vec{q} = \Phi(q)$$

2. All documents are mapped to the concept space. For each document d_i, the vector representation is then defined by:
$$\vec{d_i} = \Phi(d_i)$$

3. The score of document d_i given query q is computed by applying a vector space similarity measure. The most prominent measure is the cosine similarity:
$$\text{score}_q(d_i) = \frac{\langle \vec{q}, \vec{d_i} \rangle}{\|\vec{q}\| \|\vec{d_i}\|}$$

4. The ranking is constructed by ordering the documents according to ascending scores.

An illustration of the usage of vector space similarity in IR is given in Figure IV.3.

When implementing IR systems, comparing queries to all documents does not scale to big or large collections. In Chapter II, we mentioned the usage of inverted indexes for efficient retrieval systems based on the Bag-of-Words model. This method can be applied to concept-based retrieval as well. An inverted concept index is built by storing posting lists of documents for all concepts. For each concept, the posting list contains all documents that activate the given concept in their concept vectors. Using inverted concept indexes then allows for efficient retrieval implementations in respect to computation time and memory.

IV.2.4 Historical Overview

Since its introduction by Gabrilovich and Markovitch [2007], different application scenarios and different extensions of ESA have been proposed in literature. In the following, we will present the most prominent approaches which are also relevant in the context of this thesis. There are many more publications concerned with specific use cases and extensions of ESA. In our view, these approaches are not related to our approach and will therefore be omitted in the following list.

ESA as Semantic Relatedness Measure. In the seminal paper on ESA, Gabrilovich and Markovitch [2007] introduced their approach as a measure of semantic relatedness. Their evaluation was based on a test set of word pairs. The computed semantic relatedness of each word pair was compared to manual judgments of

human assessors. As background knowledge, they proposed to use Wikipedia articles as well as Web sites listed in the Open Directory Project (DMOZ)[3]. When using the Wikipedia database as concept source, several preprocessing steps were applied. Articles were first filtered — selecting a set of articles containing a sufficient amount of text. Further, a priori weights were assigned to articles using information gained from incoming and outgoing pagelinks in Wikipedia.

Müller and Gurevych [2008] also applied ESA as measure of semantic relatedness. They proposed to use Wiktionary[4] as alternative background knowledge source.

ESA applied to IR. Egozi et al. [2008] proposed to apply ESA to an IR scenario. As described above, they used the cosine vector similarity to compare concept vectors of queries and documents for ranking. Their evaluation was based on a dataset from TREC and topics defined by the TREC ad-hoc retrieval task (see Section II.5 for an introduction to TREC and other IR retrieval challenges). As the results of the plain ESA model were not satisfactory, they introduced a refinement step in which the weights of the concept vectors are adjusted. Along with the principles of Pseudo Relevance Feedback, the results of a first probabilistic retrieval step, based on term vector representations of documents and topics, are used to increase or decrease the weights of matching concepts in the concept representation of the topic. The refined topic concept vector is than matched against the concept representations of documents in a second step — leading to a significant improvement of retrieval results.

Cross-lingual Explicit Semantic Analysis. In 2008, we proposed an extension of ESA to multilingual settings [Sorg and Cimiano, 2008a]. Concepts in different languages that are aligned across languages can be used to define an interlingual concept space. Aligned concepts are thereby mapped to the same interlingual concept. The mapping of language-specific concept vectors of documents and topics to the interlingual concept space allows to apply ESA in Cross-lingual IR (CLIR) and Multilingual IR (MLIR) scenarios. More details will be given in Section IV.3.

An example for such aligned concepts are Wikipedia articles. They are part of the Wikipedia in a specific language but are linked to corresponding articles in other languages by cross-language links. In this chapter, we will interpret these links as concept alignments.

Potthast et al. [2008] suggested a similar approach to define CL-ESA, which was developed independently of our approach. Similar to the experiments presented in Section IV.5, they evaluated their approach on a parallel corpus using documents from the JRC-Acquis dataset. However they did not investigate the impact of the different design choices we introduce in the next section. By using Wikipedia categories to define concepts, we further extend the CL-ESA model by abstracting from single articles.

[3] `http://www.dmoz.org/` (last accessed April 8, 2011)

[4] `http://www.wiktionary.org/` (last accessed April 8, 2011)

ESA defined on Wikipedia Categories. Using Wikipedia categories as concepts was suggested by Liberman and Markovitch [2009]. They defined concept vectors in the concept space spanned by categories and applied ESA on this space to compute the semantic similarity of terms. In the next section, we will present a different approach to exploit the category structure of Wikipedia for ESA and will also apply this model to MLIR.

IV.3 Definition of CL-ESA

In this section, we present different extensions and variations of the ESA model. Our models are all based on Wikipedia as background knowledge because all the features our models depend on are present in the Wikipedia database.

Firstly, we define Cross-lingual Explicit Semantic Analysis (CL-ESA) which allows to apply ESA to documents in different languages. This requires an abstraction of single articles and categories. Links across languages in Wikipedia can be used to define equivalence classes which will be used to define interlingual concepts.

Secondly, we will present different variations of the ESA model. Many of these variations can be applied to CL-ESA as well as to monolingual ESA. A major variation is the usage of Wikipedia categories instead of articles to define the set of interlingual concepts that is then used to represent queries and documents for IR.

IV.3.1 Definition

In order to be able to use Wikipedia as multilingual background knowledge we build on the notion of *interlingual articles* $\mathcal{IA}(W)$ and *interlingual categories* $\mathcal{IC}(W)$. Thereby, W represents the union of all Wikipedia databases $W_{l_1}, \ldots, W_{l_n}$ in languages $l_1, \ldots, l_n$.

The definition of interlingual articles is based on cross-language links in Wikipedia. These directed links connect equivalent articles across languages and define the relation LL

$$\text{LL} : \mathcal{A}(W) \times \mathcal{A}(W)$$

with $(a_1, a_2) \in$ LL iff article a_1 is linked to a_2 via a cross-language link. The symmetric, reflexive and transitive closure of the LL relation can be used to define the equivalence relation $\equiv_{\text{LL}}$. The set of interlingual articles is then defined by the set of equivalence classes of all articles in different languages based on $\equiv_{\text{LL}}$:

$$\mathcal{IA}(W) := \mathcal{A}(W)/\equiv_{\text{LL}}$$

Interlingual articles can be represented by any article in the according equivalence class and therefore have representations in different languages.

Interlingual articles are thus defined by sets of equivalent articles in different languages. The language projection function Ψ_l maps interlingual articles to their

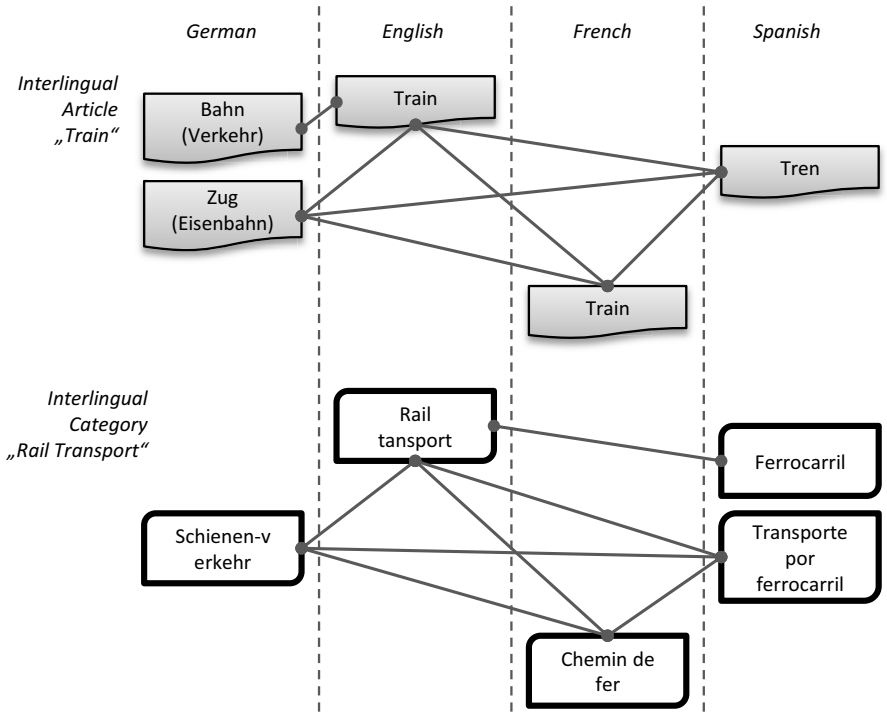


Figure IV.4: Examples for interlingual articles and interlingual categories. The connecting arrows represent cross-language links in Wikipedia. Interlingual articles (categories) correspond to equivalence classes of articles (categories) which are based on the symmetric, reflexive and transitive closure of the cross-language link relation.

Wikipedia articles in a specific language l:

$$\Psi_l : \mathcal{IA}(W) \rightarrow 2^{\mathcal{A}(W_l)}$$

Example IV.4 *In Figure IV.4 the interlingual article* `[Train]` *is visualized. It is defined by the equivalence class containing the English article* `Train` *and all articles that are connected to this article via cross-language link (and therefore defined as equivalent). In our example there are four articles that are fully connected across all languages, i.e.* `Zug` *(German),* `Train` *(English),* `Train` *(French) and* `Tren` *(Spanish). However, the German article* `Bahn (Verkehr)` *is also part of this equivalence class as it also links to* `Train`*.*

When extending ESA to cross-lingual setting, the concept space is based on interlingual articles $\mathcal{IA}(W)$. We define this extension to ESA as CL-ESA. It generalizes the ESA model presented in Equation IV.1 to documents in different languages.

For each document language l, a language specific text signature τ_l of concepts is defined:

$$\tau_l(c) := \Psi_l(c)$$

This text signature is used to compute the association strength between the interlingual concepts and documents in language l. If interlingual articles map to several articles in a specific language, the text signature is defined as the concatenation of these articles. As the concept representation is based on interlingual concepts, documents of different languages are mapped to the same concept space. Altogether, the generalized CL-ESA model for document d in language l using a tf.idf association strength function is defined as:

$$\Phi_l(d) := (\phi_l(d, c_1), \ldots, \phi_l(d, c_m))^T \tag{IV.2}$$

with

$$\phi_l(d, c) := \sum_{t \in d} \text{tf.idf}_{\tau_l(c)}(t)$$

In Section IV.4, we present ESA models based on interlingual categories instead of interlingual articles. As categories in Wikipedia are also connected via cross-language links, the definition of interlingual categories $\mathcal{IC}(W)$ is analogous to the notion of interlingual articles using the same equivalence relation. We use the same notation for the language projection function Ψ_l that maps interlingual categories to their Wikipedia categories in each language:

$$\Psi_l : \mathcal{IC}(W) \to 2^{\mathcal{C}(W_l)}$$

In the case of CL-ESA, the concept space C is then defined by the interlingual categories $\mathcal{IC}(W)$.

Example IV.5 *As an example, the interlingual category* `[Rail transport]` *is visualized in Figure IV.4. By analogy to the definition of interlingual articles, the presented equivalence class of categories contains four fully connected categories and an additional Spanish category* `Ferrocarril` *that links to the English category* `Rail transport`.

Ideally, each interlingual article (category) contains exactly one article (category) in each language, and therefore in each Wikipedia database W_l. However, this can not be assumed in the general case, which is shown by the counter examples presented in Figure IV.4. Statistics about the used Wikipedia databases, for example the average number of articles (categories) in each interlingual article (category), will be presented in Section IV.5.

IV.3.2 CL-ESA applied to CLIR/MLIR

ESA has been applied to IR by mapping both queries and documents to the concept space and then ranking documents by the vector distance to the query using the cosine distance measure. As the representations delivered by CL-ESA are language

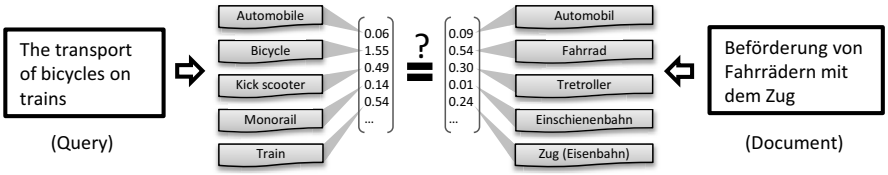


Figure IV.5: The basic principle of CL-ESA: Queries and documents of different languages are mapped to the interlingual concept space using language specific concept descriptions. Relevance of documents to queries is then measured by using similarity measures defined in the concept space.

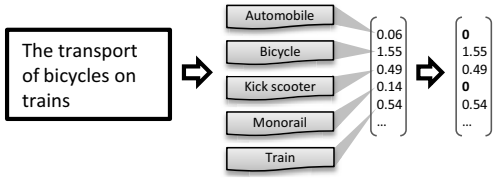


Figure IV.6: Example for pruning of ESA vectors. Dimensions having the lowest association strength values are set to zero.

independent, the framework can be naturally extended to multilingual retrieval settings without any further modification. Coming back to our running example, we interpret the English text as query, the German text as document. Figure IV.5 visualizes the CL-ESA representation of both text snippets. For retrieval purposes, the vector-based similarity of the concept vector of the query and every document in the collection can be calculated and the documents can be ranked according to this similarity.

A common technique used in all ESA implementations known to us is the dimension projection of ESA vectors. This has two main reasons: Firstly, it allows for more efficient implementations of the retrieval step due to more compact vector representations. This is also highly relevant for reducing the storage size of inverted concept indexes that are required for real-time retrieval. Secondly, the projection of the vectors also reduces noise in concept representations. The dimension projection function Π is defined as mapping function of vectors in the concept space:

$$\Pi : \mathbb{R}^{|\mathcal{IA}(W)|} \rightarrow \mathbb{R}^{|\mathcal{IA}(W)|}$$

Experiments using different functions Π will be presented in Section IV.5.

Example IV.6 *An example for the projection of CL-ESA vectors is given in Figure IV.6. Setting selected values of the CL-ESA vector to zero raises the sparseness of the vector and allows for more efficient implementations in regard to memory and storage consumption.*

	English	*German*	*French*
Query	Scary Movies	Horrorfilme	Les films d'épouvante
	Top 10 Wikipedia articles		
1	Scary Movie	Horror	La Plus Longue Nuit du diable
2	Horror	Audition	Barbara Steele
3	Scary Movie 3	Dark Water	Danger planetaire
4	Kazuo Umezu	Candyman	James Wan
5	James L. Venable	Prophezeiung (1979)	Dracula, mort et heureux de l'être
6	Horror and terror	Wolfen (Horrorfilm)	Seizure
7	Regina Hall	Alienkiller	Danvers (Massachusetts)
8	Little Shop of Horrors	Brotherhood of Blood	Fog (film,1980)
9	The Amityville Horror	Lionel Atwill	The Grudge
10	Dimension Films	Doctor X	La Revanche de Freddy

Table IV.1: The top-10 activated Wikipedia articles in the ESA vector of the example query *Scare Movies* and its translations in the three languages German, English and French.

	English	*German* $\rightarrow$ *English*
Query	Scary Movies	Horrorfilme
	Top 10 Wikipedia articles	
1	Scary Movie	Horror
2	Horror	Audition (disambiguation)
3	Scary Movie 3	Dark Water
4	Kazuo Umezu	Candyman
5	James L. Venable	Splatter film
6	Horror and terror	Prophecy (film)
7	Regina Hall	Wolfen (film)
8	Little Shop of Horrors	The Borrower
9	The Amityville Horror	Brotherhood of Blood
10	Dimension Films	Lionel Atwill

Table IV.2: The top-10 activated Wikipedia articles in the ESA vector of the example query after mapping German articles into the English Wikipedia space.

The final ranking function for a document d in language l_d given a query q in language l_q is then defined as:

$$\text{rel}(q, d) = \text{rel}(\Pi(\Phi_{l_q}(q)), \Pi(\Phi_{l_d}(d))) \tag{IV.3}$$

First, the query and the document are mapped to the concept space using the language-specific CL-ESA functions Φ_{l_q} and Φ_{l_d} (compare Figure IV.5). Both resulting concept vectors are then projected using the function Π (compare Figure IV.6). Finally, the similarity of both vectors is computed using the function rel. The output can be used to compute a relevance score between query q and document d.

	English	*French → English*
Query	Scary Movies	Les films d'épouvante
Top 10 Wikipedia articles		
1	Scary Movie	The Grudge
2	Horror	The Devils Nightmare
3	Scary Movie 3	Barbara Steele
4	Kazuo Umezu	The Blob
5	James L. Venable	James Wan
6	Horror and terror	Dead and Loving It
7	Regina Hall	Seizure (film)
8	Little Shop of Horrors	Danvers, Massachusetts
9	The Amityville Horror	The Fog
10	Dimension Films	A Nightmare on Elm Street 2

Table IV.3: The top-10 activated Wikipedia articles in the ESA vector of the example query after mapping French articles into the English Wikipedia space.

Article	*English*	*German → English*
Position in ranked ESA vector		
Scary Movie	1	555
Horror	2	1
Scary Movie 3	3	288
Scary Movie 2	4	619
The Amityville Horror (1979 film)	10	262
Scary Movie 4	12	332
Horror film	15	15
Horrorpunk	16	353
Jon Abrahams	23	235
Poltergeist (film series)	29	542

Table IV.4: Rank position of the top activated common articles that are activated both in the English and German concept vector.

IV.3.3 Example for CL-ESA

As further example in this chapter, we present the query *Scary Movies*[5] from the CLEF 2008 ad-hoc retrieval dataset, where our system performed remarkably well.

Table IV.1 contains the top-10 activated articles of the German, English and French Wikipedia in the ESA vectors of the according translations of the example query. Thereby, activation means that the corresponding dimension of an article has a non-zero value in the concept vector. The articles are ranked by the ascending value of their association strength to the query. These articles clearly differ between the languages. It is in particular interesting to observe that many results are actually named entities which differ between languages. This could be explained by the different cultural background that implies a different popularity and usage of specific named entities. Consequently, the CL-ESA vectors for the same query in different languages vary substantially, which is less optimal in a cross-lingual retrieval setting.

Table IV.2 and IV.3 contain the mappings of the German and French concept vectors to the English Wikipedia space. These mappings show that many of the top ranked articles using the English query and English Wikipedia are also present in the concept vectors that are based on queries in other languages and are then mapped to the English concept space. It can also been observed that the mapping of concept vectors introduces new concepts to the top-10 activated concepts. The reason for this is that in some cases several articles in the source Wikipedia are mapped to the same target article in the English Wikipedia. For example, when mapping the German concept vector to English, the concept *Splatter film* is present in the top ranked articles, which is not the case in the German vector. Its association score is accumulated from the score of several German concepts that are all linked to *Splatter film*.

To illustrate the actual overlap of the ESA vectors, Table IV.4 contains a list of articles that are both activated i) in the English ESA vector and ii) in the German ESA vector mapped to the English concept space. These matches are ranked according to the descending values of the association score in the English ESA vector. The positions of these matches show that the English vector and the mapped German vector have common non-zero dimensions, but the rank of these dimensions differs substantially. In an ideal setting, these ranks should be equal in both vectors. However, given the good retrieval results we achieved on this query, this indicates that these variations in the resulting CL-ESA vectors do not influence the overall retrieval performance.

IV.4 Design Choices

When instantiating the CL-ESA model, we are faced with many design choices. Particular design choices are:

[5]Document ID: `10.2452/460-AH`

- How is the concept space defined?
- How does this definition exploit the structure of Wikipedia?
- Which function should be used to compute the association strength?
- What is the best strategy for projecting ESA vectors to more efficient representations?
- How is relevance between query and document vectors modeled?

In this section, we present various alternatives for the above mentioned design choices.

Applied to the CL-ESA model as presented in Section IV.3, the different design choices essentially determine how the CL-ESA vector of queries and documents is built (see Equation IV.2) and how scores of documents given queries are computed based on these CL-ESA vectors (see Equation IV.3). Wikipedia also allows variations in the definition of the interlingual concept space, for example based on articles or on categories. In the following, we present the design choices of the generalized CL-ESA model, which can be summarized as follows:

- **Dimension Projection Function** Π: This function is used to project the CL-ESA vector to reduce the number of non-zero dimensions. When implementing ESA, processing and storing of concept vectors using all dimensions with association strength greater than 0 implies high computation and storage costs. Therefore, we project the vectors to obtain sparser representations.
- **Association Strength Function** ϕ_l: Quantifies the degree of association between document d in language l and concept c. This is based on the text signature τ_l of concept c and the text of document d.
- **Relevance Function / Retrieval Model** rel: Defines the relevance of a document to a given query on the basis of CL-ESA vector representations. While the cosine (thus assuming a geometric retrieval model) has been used as relevance function, other alternatives inspired by different approaches to monolingual IR are possible here.
- **Concept Space**: The concept space essentially defines a set of interlingual concepts as well as their textual signatures in all relevant languages. Articles and categories of Wikipedia can be used to define different interlingual concept spaces. We also present an approach that is based on the category taxonomy and uses the sub-category relation in Wikipedia to define concepts.

In this section, we will discuss particular implementations of the above functions and design choices for the concept space that can be used to instantiate the proposed generic retrieval framework based on CL-ESA. In Section IV.5, we will also provide experimental evaluation of particular instantiations applied to CLIR and MLIR scenarios.

IV.4.1 Dimension Projection

The dimension projection function is used to reduce the number of non-zero entries in each concept vector — speeding up the computation of any relevance function. We present different design choices for the realization of the dimension projection function Π that have been considered in previous literature, but never been analyzed nor compared systematically.

In the following we will use $\vec{d} = \Phi_l(d)$ as the concept vector representation of document d in language l. Let $\vec{d}_i$ denote the i-th dimension of the ESA vector of d representing association strength of d to concept c_i. The function α_d defines an order on the indices of the dimensions according to descending values such that $\forall i, j : \; i < j \rightarrow \vec{d}_{\alpha(i)} \geq \vec{d}_{\alpha(j)}$, for example $\vec{d}_{\alpha(10)}$ is the 10-th highest value of $\vec{d}$. We consider the following variants for the dimension projection function:

- **Absolute**: with $\Pi^m_{abs}(\vec{d})$ being the projected vector by restricting $\vec{d}$ to the m dimensions with highest values:, i.e. $\alpha(1), \ldots, \alpha(m)$ (as in [Gabrilovich, 2006] and [Sorg and Cimiano, 2008a])

- **Absolute Threshold**: with $\Pi^t_{thres}(\vec{d})$ being the projected vector by restricting $\vec{d}$ using threshold value t to the dimensions j with values $\vec{d}_j \geq t$ (as in [Müller and Gurevych, 2008])

- **Relative Threshold**: with $\Pi^t_{rel}(\vec{d})$ being the projected vector by restricting $\vec{d}$ to the dimensions j with values $\vec{d}_j \geq t \cdot \vec{d}_{\alpha(1)}$, $t \in [0..1]$, thus restricting it to those values above a certain fraction of the highest-valued dimension

- **Sliding Window**: This function moves a windows of fixed size l over the sorted values until the difference of the first and last value in the window falls below a threshold t. The projection cuts off all values after this position. This function was used in the original ESA model [Gabrilovich, 2006]. Formally, the projected vector $\Pi^{t,l}_{window}(\vec{d})$ is defined by restricting $\vec{d}$ to the first i dimensions according to the order α_d for which the following condition holds:

  $$\vec{d}_{\alpha(i-l)} - \vec{d}_{\alpha(i)} \geq t \cdot \vec{d}_{\alpha(1)}$$

  with threshold $t \in [0..1]$.

A relevant question is certainly how to set the parameters m and t. We address this in the experiments by first fixing a reasonable value for m in Π^m_{abs}. In order to be able to compare the different approaches, we choose the parameter t in such a way that the number of non-zero dimensions of the projected ESA vectors of all documents in the datasets amounts to m on average. The parameter l was set to 100 as in [Gabrilovich and Markovitch, 2007].

IV.4.2 Association Strength

Language specific functions $\phi_l(d, c)$ are used to calculate the value of the ESA vector for a document and the dimension corresponding to a concept c. These functions are based on the term representation of d and the text signature $\tau_l(c)$ of c in language l.

Let $|C|$ denote the number of articles, $|\tau_l(c)|$ the number of tokens in the text signature $\tau_l(c)$ of concept c. Let further $\text{tf}_d(t)$ ($\text{tf}_{\tau_l(c)}(t)$) denote the term frequency of t in document d (text signature $\tau_l(c)$) and $\text{rtf}_{\tau_l(c)}(t) = \frac{\text{tf}_{\tau_l(c)}(t)}{|\tau_l(c)|}$ represent the relative term frequency. $\text{cf}(t)$ is the number of concepts containing term t in their text signature. The inverse concept frequency is then defined as $\text{icf}(t) = \log \frac{|C|}{\text{cf}(t)}$. We can use the following functions here:

- **TFICF**: The most widely used version of the tf.idf function applied to concepts:
$$\phi_{\text{tf.icf},l}(d, c) := \sum_{t \in d} \text{tf}_d(t)\ \text{rtf}_{\tau_l(c)}(t)\ \text{icf}(t)$$
- **TFICF***: A modified tf.icf version ignoring how often the terms occur in document d:
$$\phi_{\text{tf.icf}^*,l}(d, c) = \sum_{t \in d} \text{rtf}_{\tau_l(c)}(t)\ \text{icf}(t)$$
- **TF**: An association function only based on term frequencies (ignoring inverse document frequencies):
$$\phi_{\text{tf},l}(d, c) = \sum_{t \in d} \text{tf}_d(t)\ \text{rtf}_{\tau_l(c)}(t)$$
- The **BM25** ranking function as defined by Robertson and Walker [1994] with parameters set to the following standard value: $k_1 = 2, b = 0.75$:
$$\phi_{\text{BM25},l}(d, c) = \sum_{t \in d} \frac{\text{tf}_{\tau_l(c)}(t)(k_1 + 1)}{k_1 \left((1 - b) + b \frac{|\tau_l(c)|}{\sum_{c' \in C} \frac{|\tau_l(c')|}{|C|}} \right) + \text{tf}_{\tau_l(c)}(t)} \text{icf}_{\text{BM25}}(t)$$
with
$$\text{icf}_{\text{BM25}}(t) = \log \frac{|C| - \text{cf}(t) + 0.5}{\text{cf}(t) + 0.5}$$
- The **Cosine** similarity between the tf and tf.icf vectors of d and c_l:
$$\begin{aligned} \vec{d} &= (\text{tf}_d(t_1), \text{tf}_d(t_2), \dots)^T \\ \vec{c_l} &= (\text{tf.icf}_{\tau_l(c)}(w_1), \text{tf.icf}_{\tau_l(c)}(w_2), \dots)^T \\ \phi_{\cos}(d, c) &= \frac{\langle \vec{d}, \vec{c_l} \rangle}{\|\vec{d}\| \|\vec{c_l}\|} \end{aligned}$$

Note that we have also experimented with other variations of the above presented functions, where the $\text{tf}_{\tau_l(c)}$ instead of $\text{rtf}_{\tau_l(c)}$ values were used. This resulted in a performance degradation of about 75% in all cases. For this reason, we do not present the results with the $\text{tf}_{\tau_l(c)}$ versions of the above functions in detail.

For MLIR settings, we also performed experiments putting more weight on the icf factor:

- **TFICF**^{2}: tf.idf as presented above with quadratic icf factor:

$$\phi_{\text{tf.icf}^2,l}(d,c) := \sum_{t \in d} \text{tf}_d(t)\ \text{rtf}_{\tau_l(c)}(t)\ \text{icf}(t)^2$$

- **TFICF**^{3}: tf.idf with cubic icf factor:

$$\phi_{\text{tf.icf}^3,l}(d,c) := \sum_{t \in d} \text{tf}_d(t)\ \text{rtf}_{\tau_l(c)}(t)\ \text{icf}(t)^3$$

The quadratic and cubic ICF factor gives higher weight to seldom words and lower weights to frequent words. Our experiments indeed confirm that considering ICF to the power of 2 and 3 is beneficial in MLIR but not in CLIR settings.

IV.4.3 Relevance Function

The relevance function $\text{rel}(q,d)$ defines the score of a document $d \in D$ in language l_d for a given query q in language l_q and is used to rank the documents in the retrieval process. In this multilingual setting, the function is defined on projected CL-ESA vectors $\vec{q} = \Pi(\Phi_{l_q}(q))$ of query q and $\vec{d} = \Pi(\Phi_{l_d}(d))$ of document d (see Section IV.3).

By analogy to the Bag-of-Words model, term statistics used in retrieval models can be generalized to the Bag-of-Concepts model. The *term frequency* of concept c in document d is defined as $\text{tf}_d(c) = \vec{d}_c$, the *document frequency* $\text{df}(c)$ corresponds to the number of documents in D with $\vec{d}_c > 0$. Based on this analogy, standard relevance functions defined for text retrieval can be applied to the Bag-of-Concepts model.

- The **Cosine** similarity of query and document vectors (used by all ESA implementations known to us):

$$\text{rel}_{\text{cos}}(q,d) = \frac{\langle \vec{q}, \vec{d} \rangle}{\|\vec{q}\| \|\vec{d}\|}$$

- **TFIDF:** The tf.idf function transfered to the Bag-of-Concepts model:

$$\begin{aligned} \text{rel}_{\text{TFIDF}}(q,d) &= \sum_{c \in C} \text{tf}_q(c)\ \text{rtf}_{d_i}(c)\ \text{idf}(c) \\ &= \sum_{c \in C} \vec{q}_c \frac{\vec{d}_c}{\sum_{c' \in C} \vec{d}_{c'}} \log \frac{|D|}{\text{df}(c)} \end{aligned}$$

- **KL-Divergence:** Many recent text retrieval systems use relevance functions based on the theory of language modeling. In order to be able to apply these approaches to our setting, we define the conditional probability of a concept c given a document or query d as follows:

$$P(c|d) := \frac{\vec{d}_c}{\sum_{c' \in C} \vec{d}_{c'}}$$

Intuitively this probability corresponds to the relative weight of concept c in respect to all other concepts based on the association strength values.

This definition of the conditional probability originates from the Bag-of-Words model and is inspired by Zhai and Lafferty [2001], who also describe how these probabilities can be used to define a ranking function based on the Kullback-Leibler (KL) divergence [Lee, 1999]. The KL divergence measures the difference between the query and the document model (leading ultimatively to the negative sign in the formula below). Transferred to our model, this results in the following retrieval function:

$$\text{rel}_{\text{KL}}(q, d) = -D_{\text{KL}}(q|d) \propto -\sum_{c \in C} P(c|q) \log P(c|d)$$

- **LM:** An alternative approach is to use the conditional probability $P(q|d)$ as relevance function. This distribution can be converted using the conditional distributions of documents given concepts, Bayes' law and the a priori probability of concepts $P(c) = \frac{\text{df}(c)}{|D|}$:

$$\begin{aligned} \text{rel}_{\text{LM}}(q, d) &= P(q\|d) = \sum_{c \in C} P(q|c) P(c|d) \\ &\propto \sum_{c \in C} \frac{P(c|q)}{P(c)} P(c|d) \end{aligned}$$

IV.4.4 Concept Spaces

Besides the design choices for dimension projection, association strength and relevance function presented above, a crucial question is how to define the concept space with respect to which documents will be indexed. In the original ESA model applied to Wikipedia, concepts correspond to single articles. For CL-ESA, we extended concept definitions by introducing interlingual concepts having language specific text signatures (see Section IV.3). These text signatures are defined by the content of articles and the structure given by categories in Wikipedia databases in different languages.

In the following, we propose two new approaches to define concepts based on Wikipedia: Cat-ESA and Tree-ESA. These are novel extensions of CL-ESA inspired

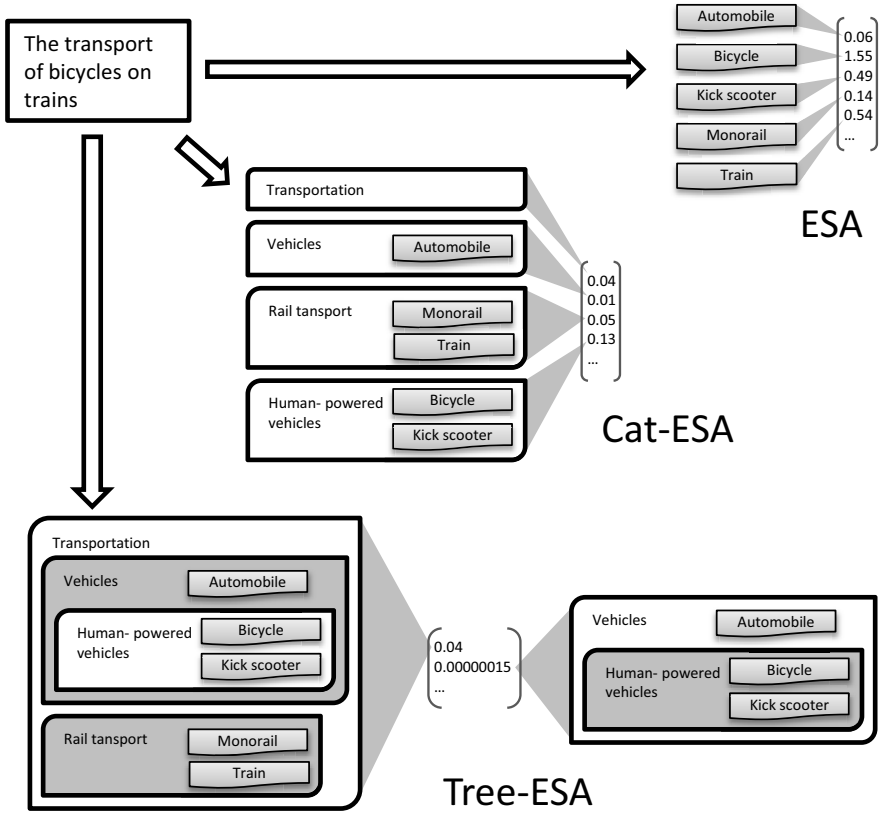


Figure IV.7: ESA vectors based on different definitions of the concept space. The original ESA model is based on articles. Concepts in Cat-ESA are defined by categories. The textual description of each category is thereby built using the articles of the category. For Tree-ESA, sub-category relations are additionally used to define the textual descriptions.

by the model introduced by Liberman and Markovitch [2009]. They presented a measure of semantic relatedness based on a concept space spanned by Wikipedia categories. In this thesis, we integrate the category based concept space in our cross-lingual ESA framework, which allows to find optimal design choices and parameters for this specific concept space. The model of Liberman and Markovitch [2009] has not been applied to IR nor CLIR/MLIR so far. Our further contribution is the evaluation of category based concept spaces in MLIR tasks. Finally, we introduce Tree-ESA that additionally exploits the hierarchical structure of categories.

Cat-ESA relies on categories of Wikipedia to define concepts. In this model, only links assigning articles to categories are considered, while relations between categories are not used.

Tree-ESA uses sub-category relations to propagate textual descriptions of concepts along the category hierarchy. Figure IV.7 contains examples for concept vectors based on different definitions of the concept space. Again, articles and categories presented in Figure IV.1 are used in this example. The ESA model defines concepts by articles, for example *Automobile*. Using Cat-ESA, concepts correspond to categories which abstract from single articles. Here, all article in the text signature of a category are also used to compute the association strength, but each article has a smaller overall weight as the text signature consists of several articles. Finally, Tree-ESA exploits the sub-category structure of Wikipedia and considers all articles along the category tree to define the text signatures.

The intuition behind these concept models is that they become more and more language-independent the more concept descriptions abstract from single articles. Therefore, indexing documents with respect to Wikipedia categories instead of Wikipedia articles might be a good choice for cross-lingual retrieval. Missing language links between articles or existing language links between articles describing different concepts may have a significant influence on the performance of CL-ESA. When using Cat-ESA with many articles in each category, these problems will surely have a smaller impact. In Tree-ESA, descriptions of categories are even bigger as they also contain subcategories. Our hypothesis is that the category-based representations used in Cat-ESA and Tree-ESA are better candidates for MLIR document models as the category structure is more stable across languages compared to the structure of articles. Our results indeed support this conclusion.

Category ESA. Category ESA (Cat-ESA) is based on a set of categories $\Gamma = \{\gamma_1, \ldots, \gamma_o\}$. We define the function

$$\text{MEMBERS} : \Gamma \rightarrow 2^C$$

that maps category γ to all articles contained in the category, which is a subset of the set of articles C.

Instantiated for Wikipedia as in our case, the categories Γ correspond to interlingual categories $\mathcal{IC}(W)$ as defined in Section IV.3. The category membership function MEMBERS is then essentially defined by category links. These links are

part of Wikipedia and assign articles to categories in a specific language l:

$$\text{CL}_l : \mathcal{A}(W_l) \rightarrow \mathcal{C}(W_l)$$

Using equivalence classes of articles and categories as defined above, these links can be generalized to interlingual articles and categories. More details about mining these links from Wikipedia will be presented in Section IV.5. As articles potentially contain more than one category link, the sets of interlingual articles of categories may not be disjoint.

In contrast to CL-ESA, the concept space of Cat-ESA is then spanned by Γ and not by C. The text signature τ of category γ is defined as the union of text signatures of all interlingual articles that are linked to one of the categories in the interlingual category:

$$\tau_{\text{Cat-ESA}}(\gamma, l) := \bigcup_{c \in \text{MEMBERS}(\gamma)} \tau_{\text{CL-ESA}}(c, l)$$

When computing term statistics, this union is equivalent to the concatenation of the articles.

Category Tree ESA. For Category Tree ESA (Tree-ESA), the categories as described for Cat-ESA are part of a tree structure. Given a single root category γ_r and a sub-category relation $\text{SUB} : \Gamma \rightarrow 2^\Gamma$, all other categories can be reached from the root category. The function $\text{TREE} : \Gamma \rightarrow 2^\Gamma$ maps a category γ to the subtree rooted in γ and is recursively defined as:

$$\text{TREE}(\gamma) := \gamma \cup \bigcup_{\gamma' \in \text{SUB}(\gamma)} \text{TREE}(\gamma')$$

As all categories are part of a tree structure without circles, this recursion stops at a leaf category node, for example

$$\text{TREE}(\gamma) := \gamma \quad \textit{if } \gamma \textit{ is a leaf}$$

Again, category links in Wikipedia, linking one category to another, can be generalized to interlingual categories and therefore be used to define the sub-category relation. The association strength of document d to category γ is then not only based on concepts in γ but also on the concepts of all subcategories. This results in the following definition of the text signature function τ:

$$\tau_{\text{Tree-ESA}}(\gamma, l) := \bigcup_{\gamma' \in \text{TREE}(\gamma)} \tau_{\text{Cat-ESA}}(\gamma', l)$$

Tree-ESA requires a tree-shaped category structure. Using Wikipedia, this is not given as some links introduce circles in the category structure. In Section IV.5, we will describe our pruning method that filters such category links to make the Wikipedia category structure usable for the proposed Tree-ESA model.

	English	*German → English*
Query	Scary Movies	Horrorfilme
Position in ranked ESA vector		
Parody films	1	55
Films	9	8
Film	10	3
B movies	24	14
Films by genre	25	4
1976 films	27	43
2000 films	33	80
Creative works	34	18
Entertainment	25	5
Teen films	38	28

Table IV.5: Rank position of the top ranked common categories that are activated both in the English and German concept vector based on Cat-ESA.

Example for Category ESA. Similar to the example presented in Section IV.3.3, we also use the query *Scary Movies* to visualize the advantages of Cat-ESA. Table IV.5 presents the ranks of categories in the Cat-ESA vectors that are activated in the English and German concept vectors of the sample topic *Scary Movies* and its German translation *Horrorfilme*.

Comparing these ranks to the results presented in Table IV.4 shows that the common categories are found at much lower rank positions. The top 10 corresponding concept in the CL-ESA vectors based on Wikipedia articles have an average rank of 320 in the German concept vector with a maximum rank of 619. Using Cat-ESA, this average is only 25 with a maximum of 80.

Our conclusion is that using categories, the resulting Cat-ESA vectors are more similar after the mapping to the interlingual concept space. This supports the hypothesis that Cat-ESA is able to compensate the differences that are present on the level of articles in the Wikipedia databases in different language. These differences do not have a strong influence on the interlingual category structure that is exploited by Cat-ESA.

IV.5 Experiments

In this section, we will present the experiments we performed to evaluate the different CL-ESA models and variants we introduced in Section IV.3. Firstly, we will introduce the different test corpora we used in the experiments. This includes parallel corpora and datasets from the CLEF evaluation campaign. Further, we present more details and statistics of the specific snapshot of the Wikipedia database that was used as background knowledge for the different CL-ESA variants. Then, we will describe the methodology of our experiments and define the different evaluation measures used.

We will present results of four different experiments. In summary, the main

objectives of our experiments are:

1. Selection of the best CL-ESA variants and parameters in respect to CLIR (Section IV.5.4). This includes the variations of the dimension projection function, the association strength function and the relevance function as presented in Section IV.4.

2. Selection of the most suitable concept space and relevance function in respect to different scenarios of cross-lingual and multilingual retrieval (Section IV.5.5). This refers to the different definitions of concept spaces presented in Section IV.4.

3. Comparison of explicit concept models (CL-ESA) to implicit concept models (Latent Semantic Indexing and Latent Dirichlet Allocation) on the task of Cross-lingual IR (Section IV.5.6).

4. Performance of CL-ESA as retrieval model applied to an international MLIR challenge. In this context, we compare our results to the results of various approaches submitted by other research groups (Section IV.5.7).

IV.5.1 Methodology and Evaluation Measures

We evaluate the presented CL-ESA models on the task of IR. The experiments are based on different multilingual datasets. Dependent on the datasets, we define two specific retrieval tasks: mate retrieval and ad-hoc retrieval (see Chapter II). These tasks differ in the type of topics and in the relevance assessments that are used to define the ground truth. For mate retrieval, documents of the dataset are used as topics and the relevance assessments can be inferred automatically. For ad-hoc retrieval, a set of topics representing specific information needs are constructed and the relevance assessments for these topics is created by humans. Both retrieval tasks will be described in detail in this section.

An important issue when presenting results in IR is also the selection of appropriate evaluation measures, as they ensure that results are comparable. Using standard measures also improves the readability, as readers usually are already familiar with these measures. In the following, we present the two different types of datasets as well as the evaluation measures used in our experiments.

Mate Retrieval. The mate retrieval task assumes the availability of a parallel corpus that can be used to evaluate cross-lingual IR approaches. We define parallel corpora as follows:

Definition IV.1 (Parallel Corpus) *A parallel corpus, supporting a set of languages $L = \{l_1, \ldots, l_n\}$, consists of a set of documents in language l_i and all the translations of each document into the other languages $l_j \in L \setminus \{l_i\}$. The translated copies of document d are called* mate documents *or* mates *of d.*

Parallel corpora in a mate retrieval setting support automatic evaluation (see Chapter II). Essentially, the retrieval task consists of retrieving the parallel document (the *mate*) as best match given a specific document as query. Per definition, the single mate in each language is thus the only relevant document for each language. Usually, mate retrieval is used to evaluate CLIR approaches in which the language of the retrieval collection is homogeneous.

We also extended mate retrieval to MLIR by i) considering for each query the translations into four languages and ii) adding the translated versions of all documents into each language to the corpus, including all mate documents of the queries. In our case, there exist thus four relevant documents for each query, i.e. the four equivalent articles or mates. This is clearly a MLIR setting as the retrieval corpus contains documents in all languages and the four mates need to be retrieved to yield a recall of 100%. Retrieving only the relevant documents in the query language — which is the query itself — will give us a recall of 25%.

Ad-hoc Retrieval Challenge. Retrieval challenges allow to compare the performance of different IR systems on a shared dataset and task. Ad-hoc retrieval tasks provide a corpus of documents and define a set of topics. Participating groups are then able to submit the retrieval results on the dataset given the topics. Usually, the top ranked documents of all groups are then pooled and judged w.r.t. their relevance to the query by human assessors.

The datasets used for ad-hoc retrieval challenges usually contain a large amount of documents — often 1,000,000 documents or more. Participating groups have therefore to ensure that the retrieval systems they develop can scale to this large amount of data.

Topics consist of *query terms*, a *description* including more details about the topic, and occasionally a *narrative* that contains even more details. At prominent IR challenges like TREC or CLEF, the topics are hand-crafted after the careful analysis of the document corpus. In the context of a specific retrieval challenge, usually 50 to 60 topics are defined. Examples of such queries are given in Section IV.5.7.

The relevance assessments acquired after assessing the documents in the result pool are often published as ground truth for the set of topics. This allows to evaluate runs that have not been part of the original result pool. Assuming a great diversity of retrieval systems that contribute to the result pool, most of the retrieved documents of new runs will be covered by the ground truth. Additionally, evaluation measures like *BPREF* that are robust in respect to non-complete relevance assessments can be used to minimize the influence of the missing assessments (see Section V.5 for a definition and applications of BPREF).

Evaluation Measures. Results of our experiments are presented using standard evaluation measures based on the ground truth, which is defined by the mate retrieval scenario. In particular, we use the following measures given the set of queries Q:

Recall at cutoff rank k (R@k): Recall defines the number of relevant documents that were retrieved in relation to the total number of relevant documents. Recall at a specified cutoff rank k is defined by only considering the top k results.

In the special case of mate retrieval, R@k defines the number of queries for which the mate document was found in the top k results. In the multilingual mate retrieval setting, it measures how many of all translations have been found.

Mean Reciprocal Rank (MRR): MRR measures the average position of the first relevant document. In mate retrieval, this corresponds to the rank of the only mate document. In contrast to R@k, this also takes into account the position of the relevant document, resulting in higher values the lower the rank of the first relevant document in the ranked result list. MRR is defined as follows:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\min_k \{d_{q,k} \,|\, \text{REL}(d_{q,k}, q) = 1\}}$$

with $d_{q,k}$ as the retrieved document given query q at rank k and $\text{REL}(d, q)$ as the binary relevance function of document d given query q.

Mean Average Precision (MAP): MAP is another standard measure in IR that is also sensitive to the rank of relevant documents. It is defined as the mean of Average Precision (AP) over all topics. AP averages precision measured at the rank of each relevant document for a given topic. Altogether, MAP is defined as:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_{k=1}^{n} \text{P@}k \cdot \text{REL}(d_{q,k}, q)}{|\{d \in D \,|\, \text{REL}(d, q) = 1\}}$$

with P@k as Precision at cutoff rank k.

As for MRR, this measure is inverse proportional to the position of the mate in the ranked retrieval list. For ad-hoc retrieval challenges, MAP has emerged as standard evaluation measure.

IV.5.2 Test Datasets

We used three different datasets in the experiments. The first two, Multext and JRC-Acquis, are parallel corpora that are used for mate retrieval. The third one, the TEL dataset, was published as corpus for an ad-hoc retrieval challenge. In the following, we describe these datasets in detail.

Multext and JRC-Acquis Datasets. We use the following two parallel corpora in our experiments:

```
<div type=RECORD id="FXAC93006ENC.0001.01.00">
  ...
  <div type="Q">
    ...
    Subject: The staffing in the Commission of the
    European Communities
    ...
    Can the Commission say:
    1.  how many temporary officials are working at
        the Commission?
    2.  who they are and what criteria were used in
        selecting them?
    ...
  </div>
  <div type="R">
    ...
    1 and 2.  The Commission will send tables showing
    the number of temporary staff working for the
    Commission directly to the Honourable Member and
    to Parliament's Secretariat.
    ...
  </div>
</div>
```

Figure IV.8: Example record of the Multext dataset.

- Multext[6] consisting of 3,152 question/answer pairs from the Official Journal of European Community (JOC).
- JRCAcquis[7] consisting of 7,745 legislative documents of the European Union.

Both corpora contain manually translated equivalents of each document in English, German, French and Spanish. In our experiments, we applied a preprocessing pipeline as commonly used in IR systems consisting of stop word removal, normalization and stemming (see Section II.1). In particular, we first eliminated stop words (for example *and* or *the*) and extremely short terms (length < 3). Then we substituted special characters to get a more consistent representation of terms (for example $\ddot{a} \rightarrow a$ or $\acute{e} \rightarrow e$). Finally we applied stemming to all words using the Snowball stemmer in the according languages.

Figure IV.8 contains the English version of a sample record of the Multext dataset, which is also available in all other languages. Each document consists of

[6] `http://aune.lpl.univ-aix.fr/projects/MULTEXT/` (last accessed April 8, 2011)
[7] `http://langtech.jrc.it/JRC-Acquis.html` (last accessed April 8, 2011)

Record	*Title or Subject*	*Annotation Terms*
1	Strength, fracture and complexity: an international journal.	Fracture mechanics, Strength of materials
2	Studies in the anthropology of North American indians series.	-
3	Lehrbuch des Schachspiels und Einführung in die Problemkunst.	Chess

Table IV.6: Example records of the TEL dataset.

Field	*Description*	*BL*	*ONB*	*BNF*
title	The title of the document	1	.95	1.05
subject	Keyword list of contained subjects	2.22	3.06	0.71
alternative	Alternative title	.11	.50	0
abstract	Abstract of the document	.002	.004	0

Table IV.7: Average frequency of content fields of the TEL library catalog records. Each record may contain several fields of the same type.

questions posted to the European Commission and the answers to these questions.

TEL Dataset. The TEL dataset was provided by the European Library in the context of the CLEF 2008/2009 ad-hoc track. This dataset consists of library catalog records of three libraries: the British Library (BL) with 1,000,100 records, the Austrian National Library (ONB) with 869,353 records and the Bibliothèque Nationale de France (BNF) with 1,000,100 records. While the BL dataset contains a majority of English records, the ONB dataset of German records and the BNF dataset of French records, all collections also contain records in multiple languages.

All of these records consist of content information together with meta information about the publication. The title of the record is the only content information that is available for all records. Some records additionally contain some annotation terms.

This dataset is challenging for IR tasks in different ways. Firstly, the text of the records is very short, only a few words for most records. Secondly, the dataset consists of records in different languages and retrieval methods need to consider relevant documents in all of these languages.

Table IV.6 shows the content information of some records of the BL dataset (the English part of TEL). As can be seen in these examples, each record consists of fields which again may be of different languages. Not all of these fields describe the content of the record but contain also meta data such as the publisher name or year of publication.

As the CLEF topics are only targeted at the content of records, we first identi-

BL			ONB			BNF		
Lang	*Tag*	*Det*	*Lang*	*Tag*	*Det*	*Lang*	*Tag*	*Det*
English	61.8%	76.7%	German	69.6%	80.9%	French	56.4%	77.6%
French	5.3%	4.0%	English	11.9%	8.0%	English	12.9%	8.2%
German	4.1%	2.9%	French	2.8%	2.1%	German	4.1%	3.8%
Spanish	3.1%	2.0%	Italian	1.8%	1.5%	Italian	2.3%	1.4%
Russian	2.7%	1.7%	Esperanto	1.5%	1.5%	Spanish	2.0%	1.4%

Table IV.8: Distribution of the five most frequent languages in each dataset, based on the language tags (Tag) and on the language detection model (Det).

fied all potential content fields. Table IV.7 contains a list of the selected fields and the average count of each field for a record. Further, we reduced additional noise by removing non-content terms like constant prefix or suffix terms from fields, for example the prefix term *Summary* in abstract fields.

In order to be able to use the library catalog records as multilingual documents, we also had to determine the language of each field. Our language detection approach is first based on the language tags provided in the dataset. These tags are present at all records of the BL dataset, for 90% of the ONB dataset and for 82% of the BNF dataset. The language distribution in the different datasets based on the language tags are presented in Table IV.8 in the column *Tag*.

However, there are several problems with language tags in the TEL dataset. Our analysis of the datasets showed that relying merely on the language tags introduces many errors in the language assignment. Firstly, there are records tagged with the wrong language. Secondly, as there is only one tag per record, language detection based on tags is not adequate for records containing fields in different languages. We therefore applied a *language classifier* to determine the language of each field of the records in the TEL dataset.

In order to identify the language for each field, we exploit a language detection approach based on character n-grams models. The probability distributions for character sequences of the size n are used to classify text into a set of languages. We used a classifier provided by the Ling Pipe Identification Tool[8] which was trained on corpora in different languages. We used the Leipzig Corpora Collection[9] that contains texts collected from the Web and newspapers and the JRC-Acquis dataset that consists of documents published by the European Union and their translations into various languages as training data.

We conducted multiple tests in order to verify the effectiveness of the language detection model. The results showed that using a 5-gram model and a 100,000 character training leads to optimal results. The classifier achieves high performance of more than 97% accuracy for text containing more than 32 characters. As this is the case for most fields in the TEL dataset, this classifier is applicable for the language detection task in our framework.

[8] `http://alias-i.com/lingpipe/` (last accessed April 8, 2011)

[9] `http://corpora.uni-leipzig.de/` (last accessed April 8, 2011)

Our language detection model determines the language for each field based on evidence from tags and from text based classification. Table IV.8 contains the language distribution in the TEL datasets based on the detection model in column *Det*. There are significant differences compared to the language assignment using only language tags, which clearly motivates the application of the language detection model. This step will probably also help to improve results in the retrieval task, as the retrieval models rely on correct language assignments.

IV.5.3 Reference Corpus

As described in Section IV.3, we used Wikipedia as background knowledge for the CL-ESA model. In particular, we used different dumps of the English, German, French and Spanish Wikipedia database.

For the experiments analyzing ESA implementation variants and the comparison to intrinsic models based on LSI or LDA, we used Wikipedia database dumps from June 2008 in the languages English, German and French.[10] As we rely on the language links to define interlingual concepts, we only chose articles that are linked across all three languages. This means that for any article a_l in language $l \in \{\text{EN}, \text{DE}, \text{FR}\}$, there exist articles $a_{l'}$ and $a_{l''}$ in the other languages that are linked to a_l. Additionally, these articles $a_{l'}$ and $a_{l''}$ are also linked to each other.

Example IV.7 *For example, given any English article a_{EN}, there is a German article a_{DE} and a French article a_{FR} that are linked to a_{EN}, and a_{DE} is also linked to a_{FR}. Other examples using existing Wikipedia articles and categories are visualized in Figure IV.4. As presented in this figure, the English article* `Train` *is linked to articles in all other languages which are again linked to each other. These articles can therefore be used to define the concept* train*. This does not hold for the German article* `Bahn (Verkehr)` *which is only linked to the English article. It is therefore not included in the definition of the interlingual concept.*

Altogether, we used 166,484 articles in each languages, as these were the articles that fulfilled our requirements.

For the experiments on concept spaces, we used Wikipedia database dumps from September 2009 in the languages English, German, French and Spanish.[11] In order to extract interlingual articles as described in Section IV.3, we filtered out articles having less than 500 characters. Then we selected interlingual articles (categories) having associated articles (categories) in three or more languages. In this case, we did not require that all of these articles (categories) are linked to each other. This resulted in 358,519 interlingual articles and 35,628 interlingual categories. 94% of the interlingual articles and 95% of the interlingual categories are

[10] `http://dumps.wikimedia.org/` (last accessed April 8, 2011)

[11] We used the most recent Wikipedia version for these experiments, which was a different dump as the one used previously. As the evaluation of the different CL-ESA variants and the evaluation of the different concept spaces are independent, this has no influence on the conclusions of our experiments.

thereby linked to exactly one article / category in each language. This approves that the Wikipedia databases are highly consistent across languages and shows the potential of Wikipedia to be used as multilingual knowledge resource and in particular to define universal (in our case called interlingual) concepts.

For Tree-ESA as described above, we also require a hierarchical link structure on interlingual articles and categories. We use category links in the different Wikipedia databases to define these links. The basic idea is to consider single category links in a specific Wikipedia as evidence for an interlingual link between the according interlingual article and interlingual category. This is defined by the *support* of an interlingual category link:

Definition IV.2 (Support of Interlingual Category Links) *The support σ of an interlingual category link between interlingual article $\alpha \in \mathcal{IA}(W)$ and interlingual category $\gamma \in \mathcal{IC}(W)$ is defined as:*

$$\sigma(\alpha, \gamma) := |\{(a, c) \in \boldsymbol{CL}_l \mid l \in L, a \in \Psi_l(\alpha), c \in \Psi_l(\gamma)\}|$$

with CL_l being the relation defined by category links in Wikipedia W_l and L being the set of supported languages. Ψ_l is the projection of interlingual articles or categories to articles or categories in Wikipedia W_l as defined in Section IV.3.

We use a support threshold of $\sigma \geq 2$ to select interlingual category links, resulting in 605,672 links between articles and categories. We apply the same definition of support and the same threshold for links between interlingual categories, resulting in 58,837 links. The intention of the selection process of interlingual category links is the reduction of noise. Links, that are only supported by one category link, are only present in a single Wikipedia database. They are missing in all other languages and might therefore be redundant or even introduce errors.

As root for the category tree used in Tree-ESA, we selected the interlingual category associated with the top category in the different language-specific Wikipedias. 86% of the interlingual categories extracted before are connected to this root by category links and are therefore part of the category tree. As the category link graph in Wikipedia contains cycles, we use a breadth-first algorithm that explores the graph starting from the root category. All links introducing circles are ignored in the exploration, thus resulting in a tree-shaped category structure.

IV.5.4 Evaluation of ESA Model Variants

As described in Section IV.3, the ESA model allows different design choices. To find best parameters for CLIR, we performed experiments testing the influence of different variants on the retrieval performance. Our experiments have been carried out in an iterative and greedy fashion in the sense that we start form the original ESA model as a baseline. Then we iteratively vary different parameters and always fix the best configuration before studying the next parameter. The sequence of optimizations is given by the nesting of functions in Equation IV.3. We first optimized the

parameters of the inner functions that do not depend on the output of other functions and then proceeded to the outer functions. At the end of our experiments, we will thus be able to assess the combined impact of the best choices on the performance of the ESA model.

In summary, the contributions of the experiments on ESA model variants are the following:

1. We identify best choices for the parameters and design variants of the CL-ESA model on a CLIR scenario.

2. We show that the CL-ESA model is sensitive to parameter settings, heavily influencing the retrieval outcome.

3. We show that the settings chosen in the original ESA model are reasonable, but can still be optimized for CLIR and MLIR settings.

Experimental Settings. Results of the evaluation of variants of the CL-ESA model are presented in Figures IV.9 to IV.12. As the observed effects were constant across measures, we only present recall at cutoff rank 1 (R@1). For experiments on the Multext corpus we used all documents (2,783) as queries to search in all documents in the other languages. The results for language pairs were averaged for both retrieval directions (for example using English documents as queries to search in the German documents and vice versa). For the JRC-Acquis dataset, we randomly chose 3,000 parallel documents as queries (to yield similar settings as in the Multext scenario) and the results were again averaged for language pairs. This task is harder compared to the experiments on the Multext corpus as the search space consists of 15,464 documents and is thus bigger by a factor of approximately 5. This explains the generally lower results on the JRC-Acquis dataset.

To prove the significance of the improvement of our best settings (projection function Π_{abs}^{10000}, association strength function tf.icf*, cosine retrieval model) we carry out paired t-tests (confidence level 0.01) comparing the best settings pairwise with all other results for all language pairs on both datasets. Results where the differences are not significant with respect to all other variants at a confidence level of 0.01 are marked with *X* in Figures IV.9 to IV.12.

In the following we discuss the results of the different variations of the CL-ESA model:

Projection Function. We first used different values for the parameter m in the projection function Π_{abs}^{m}. The results in Figure IV.9 showed that $m = 10,000$ is a good choice for both datasets.

On the basis of this result, we investigated different projection functions. In order to be able to compare them, we set the different threshold values t such that the projected ESA vectors had an average number of approx. 10,000 non-zero dimensions. An exception is the function *sliding window (orig.)* where we used the

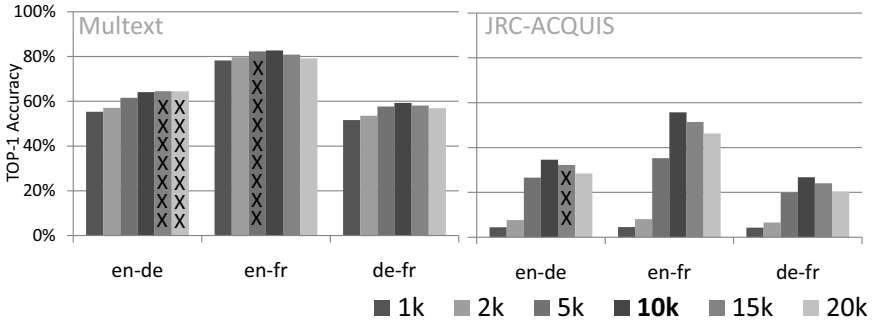


Figure IV.9: Variation of m in the projection function Π^m_{abs} using the tf.icf* association function and cosine retrieval model. Results that have no significant difference to the results of our best setting are marked with X.

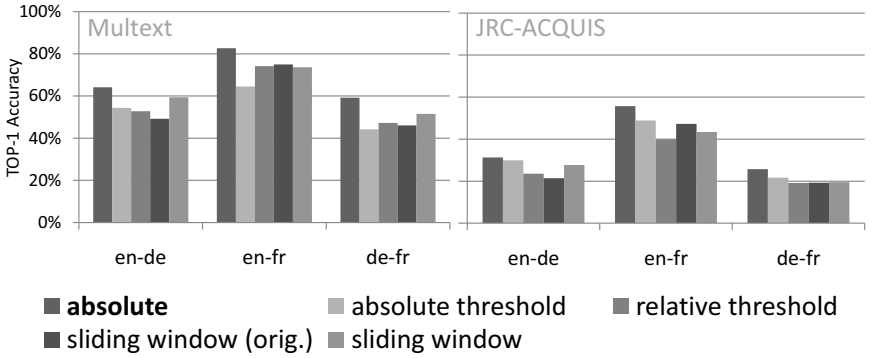


Figure IV.10: Variation of the projection function Π using the tf.icf* association function and cosine retrieval model.

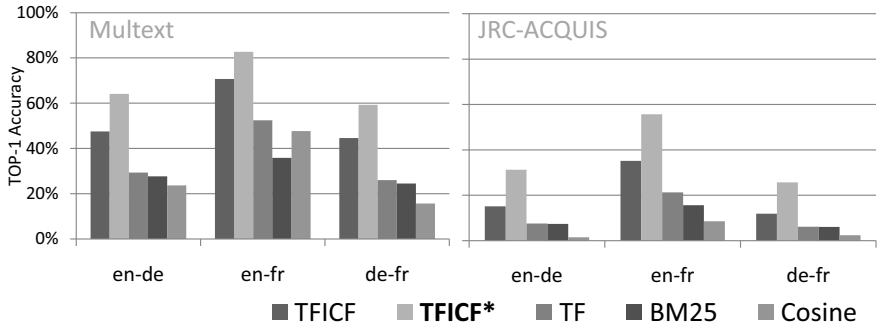


Figure IV.11: Variation of the association strength function ϕ_l using the projection function $\Pi_{abs}^{10,000}$ and cosine retrieval model.

parameters described in [Gabrilovich, 2006]: threshold $t = 0.05$ and window size $l = 100$. Using an absolute number of non-zero dimensions yielded the best results (see Figure IV.10), the difference being indeed significant with respect to all other variants. Thus, we conclude that neither the settings of the original ESA approach (sliding window) nor in the model of Gurevych et al. [2007] (fixed threshold) are ideal in our experimental settings. For the remaining experiments, we therefore use the absolute dimension projection function that selects 10,000 articles ($\Pi_{abs}^{10,000}$).

Association Strength. The results in Figure IV.11 show that the functions tf.icf (used in the original ESA model) and tf.icf* perform much better compared to the other functions. The better performance of tf.icf* which ignores the term frequencies in the queries was indeed significant w.r.t. all other alternatives for all language pairs considered on both datasets. We thus conclude that the settings in the original ESA model are reasonable, but, surprisingly, can be improved by ignoring the term frequency of the terms in the document to be indexed. The low results using the tf function show that icf is an important factor in the association strength function. Otherwise, the normalization of the tf.icf values (= cosine function) reduces the retrieval performance substantially.

Retrieval Model. Experiments with different retrieval models lead to the result that the cosine function, which is used by all ESA implementations known to us, constitutes indeed a reasonable choice. All other models perform worse (the difference being again significant for all language pairs on both datasets), which can be seen at the charts in Figure IV.12, especially on the JRC-Acquis dataset.

Discussion. Our results show on the one hand that ESA is indeed quite sensitive to certain parameters (in particular the association strength function and the retrieval

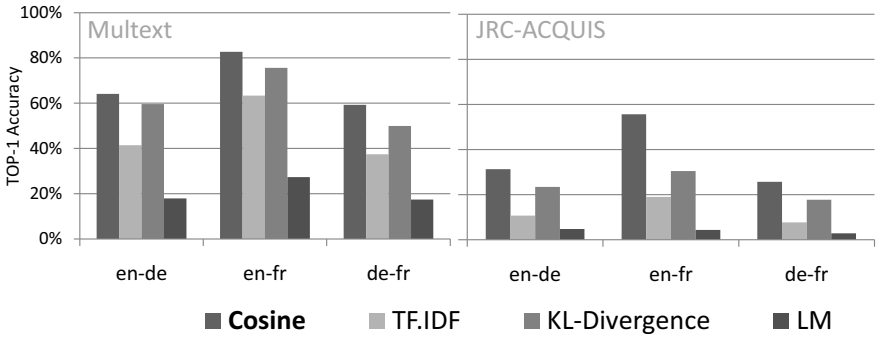


Figure IV.12: Variation of the retrieval model using $\Pi_{abs}^{10,000}$ and tf.icf*.

model). The design choices have a large impact on the performance of the ESA based retrieval approach. For example, using tf_c values instead of rtf_c values (which are length normalized) in the association strength function decreases performance by about 75%. Unexpectedly, abstracting from the number of times that a term appears in the query document (using tf.icf*) improves upon the standard tf.icf measure (which takes them into account) by 17% to 117%. We have in particular shown that all the settings that are ideal in our experiments are so indeed in a statistically significant way. The only exception is the number of non-zero dimensions taken into account, which has several optimal values.

On the other hand, the results of our experiments confirm that that the settings in the original ESA model ($\Pi_{window}^{0.05,100}$, tf.icf, cosine) [Gabrilovich and Markovitch, 2007; Gabrilovich, 2006] are reasonable. Still, when using the settings that are ideal on both datasets according to our experiments ($\Pi_{abs}^{10,000}$,tf.icf*,cosine), we achieve a relative improvement in TOP-1 accuracy between 62% (from 51.1% to 82.7%, Multext dataset, English/French) and 237% (from 9.3% to 31.3%, JRC-Acquis dataset, English/German). This shows again that the settings can have a substantial effect on the ESA model and that ESA shows the potential to be further optimized and yield even better results on the various tasks it has been applied to.

Finally, all experiments including the German datasets have worse results compared to the English/French experiments. This is likely due to the frequency of specific German compounds in the datasets, which lead to a vocabulary mismatch between documents and Wikipedia articles. However an examination of this remains for future work.

IV.5.5 Concept Spaces for Multilingual Scenarios

Two different retrieval problems have been considered in the context of retrieval systems bridging between languages: Cross-lingual IR (CLIR) and Multilingual IR

(MLIR). While CLIR is concerned with retrieval for given language pairs (*i.e.* all the documents are given in a specific language and need to be retrieved to queries in another language), MLIR is concerned with retrieval from a document collection where documents in multiple languages co-exist and need to be retrieved to a query in any language.

While CL-ESA model variants where evaluated on a CLIR scenario, we also tested its performance on multilingual settings. This is an inherently more difficult problem as documents of different languages compete in the retrieval process. We will define this problem as *language biases* in this section. One of the goals in the following experiments is to show that CL-ESA is not strongly affected by language bias. In these experiments, we particularly focus on different concept spaces that are used by CL-ESA, Cat-ESA and Tree-ESA.

In summary, the contributions of the experiments on multilingual retrieval scenarios are the following:

1. We present and analyze the performance of different CL-ESA models on a MLIR task.

2. We show that CL-ESA using concept spaces based on Wikipedia categories improves performance on MLIR settings compared to concept spaces based on articles. Our results show that there is indeed a gain in MAP from 18% to 39%.

3. We show that CL-ESA applied to MLIR is not strongly affected by two different types of language biases (which will be defined in the following).

4. We argue that CLIR and MLIR are quite different problems for which different retrieval approaches need to be applied. In particular, we show how the performance in CLIR and MLIR tasks varies substantially for the same parametric choices.

Definition of Language Bias. One of the main problems that make MLIR very challenging could be defined as *language bias*. This problem manifests itself in two facets:

- The *language bias type I* captures the intuition that, given a query in a certain language, the retrieval system is in principle biased to return documents in that specific language. An extreme case is multilingual retrieval without query translation. Most terms in the query will only be matched to documents of the same language which are then retrieved in the top ranked results. However, even if query translation is applied, information will often be lost in the translation process, for example by incomplete translations, which has a similar effect on the retrieval results.

- The *language bias type II* captures the fact that retrieval systems based on standard term weighting models are intrinsically biased to rank documents in

Concept Model	*ID*	*Retrieval Model*	*Multext* *R@10*	*Multext* *MAP*	*JRCAcquis* *R@10*	*JRCAcquis* *MAP*
Bag of Words	1	BM25	.27	.27	.25	.25
CL-ESA	2	TFICF	.27	.27	.26	.26
	3	TFICF^2	.30	^{2}.29	.28	^{2}.27
	4	TFICF^3	.33	^{3}.33	.29	^{3}.28
Cat-ESA	5	TFICF	.36	^{3}.33	.27	.27
	6	TFICF^2	.44	^{4,5}.39	.32	^{5}.30
	7	TFICF^3	.47	^{6}.43	**.36**	^{6}**.33**
Tree-ESA	10	TFICF	.36	^{3}.33	.26	.26
	11	TFICF^2	.48	^{10}.42	.30	^{3,5,10}.28
	12	TFICF^3	**.51**	^{6,11}**.46**	.33	^{5,11}.31

Table IV.9: Results of mate retrieval experiments using queries and documents of all languages. Statistically significant differences according to paired t-test at confidence level .001 are marked with the ID of the compared result.

the most prominent language of the collection relatively high. This problem is again based on the matching of query terms to documents and on the a priori probability of documents. The high share of documents in the most prominent language leads to a high probability that these documents are at least matched to some of the query terms. In contrast, this probability is much lower for the small share of documents in other languages.

Both problems arise in particular in connection with document models used in monolingual IR, which are language-specific. In the following experiments, we will analyze how CL-ESA and its variants are affected by the language bias problem. We will also examine other baseline approaches in respect to the language bias in MLIR tasks.

Experimental Settings. To define queries for our experiments, we randomly selected 50 documents each from the Multext and from the JRC-Acquis dataset. We then used all translations of these documents as queries, which resulted in 200 multilingual queries for each dataset. As evaluation measures we used Mean Average Precision (MAP) and Recall at cutoff level of 10 (R@10).

Document Models. The results on the multilingual mate retrieval task using the different CL-ESA abstractions and retrieval models are presented in Table IV.9 which shows R@10 and MAP values for the different configurations under analysis. The results clearly show that the models relying on categories as concepts instead of articles are indeed superior, with MAP improvements of 70% (Multext) and 32% (JRC-Acquis) to the BoW model baseline and 39% (Multext) and 18% (JRC-Acquis) to CL-ESA.

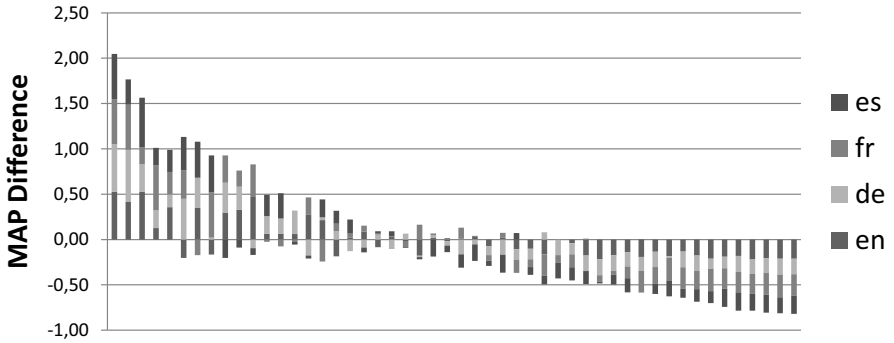


Figure IV.13: Differences in AP to MAP for each query. The results using different query languages are presented in a single additive bar. The experiments were performed on a mate retrieval setting using Tree-ESA with the TFICF^3 model on the Multext dataset.

The interlingual document representation given by Cat-ESA and Tree-ESA is therefore more suited for MLIR than any of the other models considered. Interestingly, there seems to be no significant difference between Cat-ESA and Tree-ESA. The fulfillment of our intuition that category-based concept models smooth the structural and content-based differences of Wikipedia databases in different languages is a possible explanation of these results.

When considering the different retrieval models, the results vary significantly. For all category-based models we achieve high performance gain when using tf.icf^2 instead of tf.icf on both datasets (MAP improvements between 8% to 27%). About the same improvement is observed again when using tf.icf^3 instead of tf.icf^2 (MAP improvements between 10% to 11%). All these improvements using the different retrieval models have been tested to be statistically significant at confidence level .001 using a paired t-test. Giving more weight to non-frequent terms is therefore beneficial for multilingual mate retrieval. A possible explanation is that reducing the weight of frequent terms also reduces the language bias, as this bias might be induced by the different distributions of these terms in each language.

Language Bias. The fact that the R@10 values are close to 50% for Cat-ESA and Tree-ESA shows that to a query in a certain language the system is retrieving about two mates in the top ten results on average. This shows that Cat-ESA and Tree-ESA can deal with the language bias reasonably well. The fact that the performance is comparable across languages and not biased to any specific languages can be visually inferred from the values presented in Figure IV.13, which plots Average Precision (AP) results for English, German, French and Spanish queries on the Multext dataset using Tree-ESA with tf.icf^3. The X-axis consists of the 50 queries, each having a translation in all languages. On the Y-axis, the difference of the query specific AP to

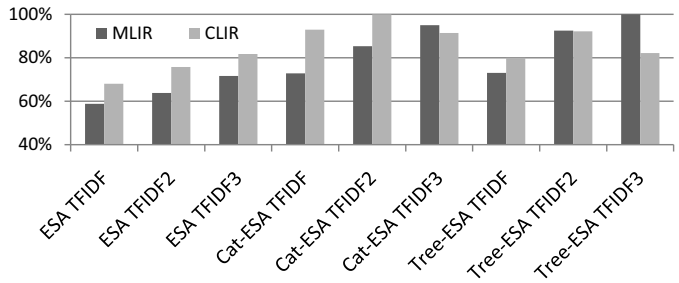


Figure IV.14: Relative performance with regard to best MAP values for mate retrieval using different CL-ESA models on multilingual documents and on English documents of the Multext dataset.

the MAP of all queries is plotted. These differences were determined for each query language and the results are presented as a single additive bar.

The results show that the performance of our approach does not vary across languages as differences to MAP are mostly consistent when using the same query in English, German, French or Spanish. For some queries, we achieve an AP above average which is observed using all query languages. Considering the bad performing queries, the results are also consistent when using queries in the different languages.

CLIR vs. MLIR In this section, we introduced MLIR as an inherently more difficult problem than CLIR. To support this claim we have additionally performed CLIR experiments on the Multext dataset. In this case the target document collection consists only of documents in one language. Experiments using Cat-ESA with the tf.icf^2 model on English documents resulted in high MAP values ranging from .72 to .85 using German, French and Spanish queries. This shows that our model is also applicable to CLIR but also that CLIR is indeed a much easier problem.

The results for CLIR presented above are based on an optimal selection of parameters. Interestingly, we appreciate that the best performing configuration is different from the best performing configuration for MLIR. To illustrate this effect we set the relative performance for both best performing configurations on the Multext dataset, on the one hand for CLIR and on the other hand for MLIR, to 100% in Figure IV.14. The performance of all other configurations is shown relatively to the best configuration. The results in Figure IV.14 show that while using cubic ICF values always improves results for MLIR, this is definitely not the case for CLIR. Using Cat-ESA and Tree-ESA, we observe a significant drop in MAP compared to the quadratic icf model. This effect is also confirmed using Tree-ESA for CLIR on the JRC-Acquis dataset. This shows that model and parameter selection has to be adjusted to the specific settings, *i.e.* CLIR or MLIR. As these are different problems, good approaches to solve one setting might not be optimal on the other.

IV.5.6 External vs. intrinsic Concept Definitions

In Section IV.5.4 and IV.5.5, we presented experiments on CL-ESA model variants as well as variations of the oncept space. However, we did not compare CL-ESA to alternative concept based approaches to CLIR. In the following, we introduce two models based on intrinsic concept definitions. Both models are evaluated in regard to retrieval performance using the mate retrieval scenario and they are compared to CL-ESA.

In summary the contributions of the experiments comparing CL-ESA to intrinsic concept models are the following:

1. Training of intrinsic models on Wikipedia results in much lower retrieval performance. CL-ESA is therefore able to exploit Wikipedia as background knowledge more effectively.

2. CL-ESA shows comparable results to intrinsic models that are trained on the dataset, using a train/test split. However, CL-ESA is not trained on the dataset but a generic model based on background knowledge.

Intrinsic Concept Definitions. For the comparison to alternative concept models, we chose LSI and LDA as representative approaches for deriving implicit (latent) concepts from a text collection.

In the monolingual case, LSI is used to identify latent topics in a text corpus. These topics, which correspond to concepts in CL-ESA, are extracted by exploiting co-occurrences of terms in documents. This is achieved by Singular Value Decomposition (SVD) of the term-document matrix [Deerwester et al., 1990]. The latent topics then correspond to the eigenvectors having the largest singular values. This also results in a mapping function from term vectors to *topic vectors*. LSI was originally used for dimensionality reduction of text representation and for improved retrieval of synonyms or similar terms. By using parallel training corpora, LSI can also been applied to CLIR such that the resulting topics span terms of different languages [Littman et al., 1998]. More details and an example of LSI is given in Chapter III.

As an instance of probabilistic latent topic models, we consider the LDA based generative model. The basic idea of this approach is to abstract from particular words and to represent multilingual documents by mixtures over a set of latent concepts (*i.e.* hidden document-specific themes of interest), whereby each concept is characterized by a fixed conditional distribution over terms from different languages. LDA assumes that all multilingual terms (both observed and previously unseen) are generated by randomly chosen latent concepts. In contrast to the SVD used in LSI, LDA has a well founded probabilistic background and tends to result in more flexible model fitting [Blei et al., 2003]. It allows resources to belong to multiple latent concepts with different degrees of confidence and offers a natural way of assigning probabilistic feature vectors to previously unseen resources.

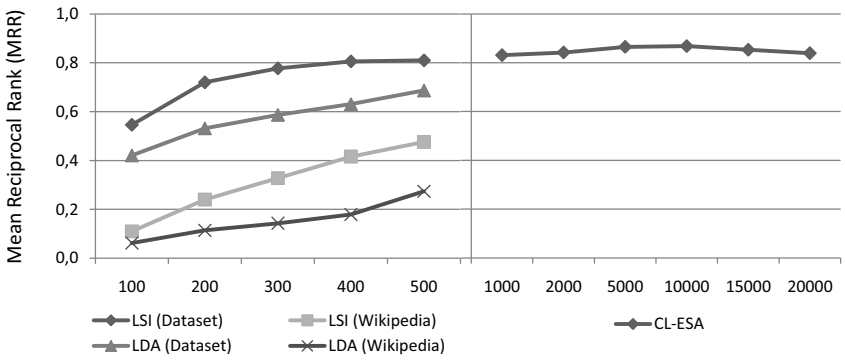
Dataset	*Method*	en-fr R@1	en-fr MRR	en-de R@1	en-de MRR	de-fr R@1	de-fr MRR
Multext	CL-ESA	**.83**	**.87**	**.72**	**.78**	**.64**	.71
	LSI (Dataset)	.71	.81	.60	.72	.59	**.72**
	LSI (Wikipedia)	.36	.48	.13	.21	.13	.22
	LDA (Dataset)	.11	.69	.04	.48	.05	.47
	LDA (Wikipedia)	.01	.27	.01	.16	.01	.14
JRC-Acquis	CL-ESA	**.56**	.61	**.35**	.40	.27	.32
	LSI (Dataset)	.52	**.65**	.29	**.45**	**.34**	**.49**
	LSI (Wikipedia)	.18	.27	.07	.12	.07	.13
	LDA (Dataset)	.08	.62	.12	.36	.04	.38
	LDA (Wikipedia)	.01	.09	.01	.07	.01	.08

Table IV.10: Results for the mate retrieval experiments on the Multext and JRC-Acquis dataset using optimal settings of the topic numbers for LSI/LDA (500) and of the non-zero dimensions for CL-ESA vectors ($\Pi_{abs}^{10,000}$). Evaluation measures are Recall at cutoff rank 1 (R@1) and Mean Reciprocal Rank (MRR).

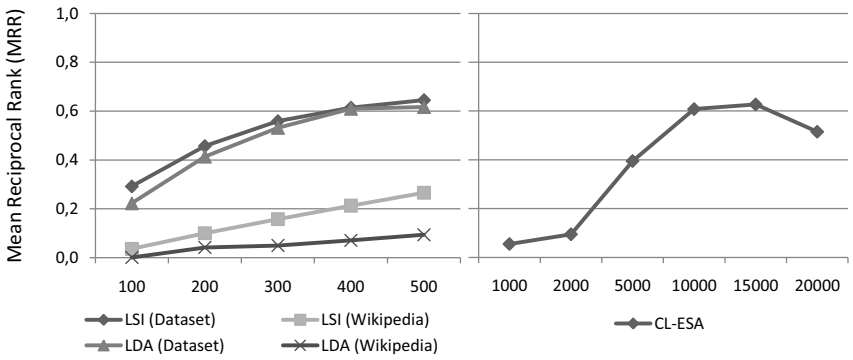
Experimental Settings. For the comparison of external vs. intrinsic concept models we used the same experimental settings as in Section IV.5.4, using all documents of the Multext dataset and 3000 randomly selected documents of the JRC-Acquis dataset as queries. We considered English, German and French as baseline languages and performed accordingly 6 series of experiments with all possible language pairs. The results for one language pair in both directions, for example English-German and German-English, were then averaged.

Concerning the LSI implementation, the Wikipedia corpus after preprocessing was still too huge and too sparse for an efficient LSI, which would require each document and its mate to have between 50 and 500 terms. Therefore, we skipped all terms which appear less than 15 times and finally used all documents containing at least 75 terms. In this way for example, the number of English-French document pairs was reduced to 54,764. These restrictions were applied to each language pair separately. Because of the sparseness of the term-document matrices, a similar process was applied to the JRC-Acquis dataset.

Results. Figure IV.15 shows sample MRR results for mate retrieval experiments between English and French documents. CL-ESA reaches its peak performance with a projection of concept vectors to 10,000 non-zero dimensions ($\Pi_{abs}^{10,000}$, see Section IV.4), which shows that generally it needs a minimum number of associated concepts to perform reasonably, but clearly also reaches a point were further associations to more concepts start introducing noise. For latent topic models, the accuracy tends to increase from 100 to 500 topics. The exploration of computationally much more expensive latent models ends with 500 topics, due to computational limitations of our servers. In the experiments we report below, we used these optimal settings



(a) Multext dataset



(b) JRC-Acquis dataset

Figure IV.15: Results for the mate retrieval experiments on English and French documents for different topic numbers for LSI/LDA and different numbers of non-zero dimensions (m parameter) for CL-ESA (tf.icf*, Π^m_{abs}).

(projected to 10,000 non-zero dimensions for CL-ESA, 500 for latent models) for all language pairs.

Table IV.10 shows the corresponding summary of achieved results. It includes R@1 and MRR of the mate documents. In addition, for latent topic methods we compared the retrieval characteristics for different choices of the background knowledge resource, namely Wikipedia vs. the test collection itself at the ratio of 60% for learning and 40% for testing. The results show that w.r.t. R@1, CL-ESA outperforms both other models on all language pairs of the Multext dataset and on the English-French and English-German pairs of the JRC-Acquis dataset, but not on German-French. With respect to MRR, LSI performs better than CL-ESA in some cases, but only when it has been trained on the retrieval document collection itself. When LSI is trained on Wikipedia as an aligned corpus, results are in all cases worse. This shows that CL-ESA is indeed superior as it does not rely on an aligned corpus to be trained on to deliver good results.

Discussion. Our results show that using Wikipedia as multilingual knowledge resource leads to significantly worse results for latent topic models (in contrast to the case when they are trained on the retrieval document collection). The explanation of this phenomenon is twofold. On the one hand, Wikipedia is not a fully parallel corpus and linked articles may show substantial variation in size, quality, and vocabulary. On the other hand, there is a serious vocabulary mismatch between Wikipedia and our thematically focused test collections. For instance, in the Multext collection, 4,713 English terms (44%), 8,055 German terms (53.8%) and 7,085 French terms (53.6%) are not covered by Wikipedia articles at all. We also assume that the performance of LDA observed in our experiments can be further improved by heuristic model tuning, including optimization of concentration parameters for Dirichlet priors, or smoothing of estimated multinomial parameters (as described in [Blei et al., 2003]).

Overall, it can be claimed that CL-ESA clearly outperforms LSI/LDA unless the latter are trained on the document collection and not on Wikipedia as a generic, domain-independent resource. The availability of aligned corpora is a serious restriction, so that CL-ESA is clearly the preferred model here as it delivers reasonable results requiring no data aligned across languages besides Wikipedia.

A further crucial advantage is its excellent scalability: CL-ESA does not require comprehensive computations with nonlinear space/time behavior and can be practically performed within 20-30 min for any desired number of topics in each of our test collections. The complexity of CL-ESA depends on the number of activated concepts k in the concept vector of the query — which was fixed to different numbers $\leq 20,000$ in our experiments using the projection function — and the length of the entries in the inverted concept index. The maximal size of these entries is given by the number of documents n, which results in the worst case complexity of $O(kn)$. On average, these entries are much smaller. Their average length depends on the number of concepts $j \gg k$ and the number of activated concepts k in each document

Field	*Content*
CLEF 2008: `10.2452/460-AH`	
Title	Scary Movies
Description	Find publications discussing the making or describing the details of any horror film.
CLEF 2009: `10.2452/701-AH`	
Title	Arctic Animals
Description	Find documents about arctic fauna species.

Table IV.11: Examples for topics used in the CLEF ad-hoc tasks.

concept vector: $\frac{kn}{j}$. This results in the average-case complexity of $O(\frac{k^2}{j}n)$.

In contrast, LSI and LDA have much higher complexity. Even with significantly lower dimensionality, the computation of the LSI and LDA models took between 3 hours and 7 days. The complexity of LSI is based on the singular value decomposition of the document-term matrix, which as the complexity of $O(n^3)$.

IV.5.7 Experiments on the CLEF Ad-hoc Task

We participated in two different retrieval challenges organized in the context of CLEF: the ad-hoc task in 2008 and the ad-hoc task in 2009. In the following, we describe these tasks and present the achieved results of our CL-ESA-based retrieval systems. Both challenges use the TEL dataset as introduced in Section IV.5.2 that contains documents from the British Library (BL), the Austrian National Library (ONB) and the Bibliothèque Nationale de France (BNF).

CLEF 2008 Ad-hoc Task. The CLEF ad-hoc TEL task was divided into monolingual and bi-lingual tasks. 50 topics in the main languages English, German and French were provided. The topics consist of two fields, a short title containing 2-4 keywords and a description of the information item of interest in terms of 1-2 sentences. An example of a topic is given in Table IV.11.

The objective is to query the selected target collection using topics in the same language (monolingual run) or topics in a different language (bi-lingual run) and to submit the results in a list ranked with respect to decreasing relevance. In line with these objectives, we submitted results of six different runs to CLEF 2008. These are the results of querying English, German and French topics to the TEL English dataset and English, German and French topics to the TEL German dataset.

The following parameter settings as described in the implementation section were used for these experiments:

ESA vector length: We used different lengths of the ESA vector to represent topics and records. For the topics we used $k = 10,000$, *i.e.* the 10,000 Wikipedia articles with the strongest association to a specific topic were used to build the

Dataset	*Topic language*	*MAP*
TEL English (BL)	English	17.7%
	German	7.6%
	French	3.6%
TEL German (ONB)	English	6.7%
	German	9.6%
	French	5.1%

Table IV.12: Retrieval results on the CLEF 2008 ad-hoc task measured by Mean Average Precision (MAP).

ESA vector for this topic. For the records, we used $k = 1,000$. The difference between the lengths is mainly due to performance issues. We were only able to process the huge amount of records by limiting the length of the ESA vectors for records to 1,000 non-zero entries. As only 50 topics were provided, we were able to use more entries for the ESA vectors for topics. Our intention thereby was to improve recall of the retrieval, which is achieved by activating more concepts in the topic vector.

Article selection: The CL-ESA model used for experiments submitted to CLEF 2008 was based on the concept space defined by all articles in the English and German Wikipedia. We did not apply any pre-selection of articles, for example based on the linkage across languages. The main problem of this setting is the loss of many dimensions in the mapping process across languages, as not all of the articles corresponding to a non-zero ESA vector entry have an existing cross-language link to the Wikipedia in the target language. In this case, the information about this dimension is lost in the mapping process. A pre-selection of articles that are used to build the concept space overcomes this problem, which was used for example in our experiments submitted to CLEF 2009.

CLEF 2008 Results. Table IV.12 contains the CLEF 2008 results of our submitted experiments measured by Mean Average Precision (MAP). The results show, that monolingual retrieval still outperforms cross-lingual retrieval. On the BL dataset, retrieval performance is more than doubled when using English topics compared to using German or French topics. On the ONB datasets, the differences are much smaller.

In addition to the submitted experiments we also conducted experiments on the TEL dataset to better quantify and understand the impact of certain parameters on the result quality. As we were not able to evaluate the results apart from the submitted ones, we decided to examine the result overlap for queries in different languages on the same dataset. This measure can be seen as a quality measure for the capability of retrieving relevant documents across languages. Ideally, queries in different

Article restriction	*Topic language pair*	*Average result overlap*
No restriction	English - German	21%
	English - French	19%
	German - French	28%
Articles with existing cross-language link	English - German	39%
	English - French	51%
	German - French	39%

Table IV.13: Result overlaps of the top retrieved documents using topics in different languages on the BL dataset. For the CL-ESA model, two different reference corpora are tested: the set of all Wikipedia articles vs. the set of articles with an existing cross-language link.

languages should result in the same set of retrieved records. We computed the result overlap for two different settings. Firstly, we used the same settings as in the submitted results. For the second set of experiments, we further restricted the Wikipedia articles that were used for ESA indexing to articles with at least one language link to one of the two other languages considered. Table IV.13 contains the result overlaps for topic pairs in different languages on the BL dataset.

The results show that we were able to substantially improve the retrieval methods according to the results overlap measure. By using the restricted set of Wikipedia articles, the retrieval results are more similar when using queries that are defined by different translations of the same topic. For example, the overlap of using the English and French queries of the same topics is raised from 19% to 51%. Our assumption is that the results on the retrieval task would also improve, but we did not manage to submit an official run on time for this challenge.

CLEF 2009 Ad-hoc Task. The CLEF 2009 ad-hoc topics were similar to the topics from CLEF 2008. The 50 topics have the same format consisting of two fields, a short title containing 2-4 keywords and a description of the information item of interest in terms of 1-2 sentences. An example topic used in CLEF 2009 is given in Table IV.11.

Again, monolingual and bi-lingual challenges were defined for the CLEF 2009 ad-hoc task based on the TEL corpus. We submitted results of six different runs, querying English, German and French topics to the BL, ONB and BNF datasets.

In contrast to the experiments submitted to CLEF 2008, we applied the language detection model as preprocessing step to all TEL records. As shown in Section IV.5, this improves the classification rate of the language of fields compared to only relying on the specified language tags. This information is then used to build language specific indexes. A lower error rate in language detection should therefore also improve the overall retrieval performance.

Topic Lang.	*Retrieval Method*	*MAP*	*P@10*	*R@100*
	BL Dataset			
en	Baseline (single index)	.35	.51	.55
	Multiple Indexes	.33	.50	.52
	Concept + Single Index	.35	.52	.54
de	Baseline (single index)	.33	.49	.53
	Multiple Indexes	.31	.48	.51
	Concept + Single Index	.33	.49	.53
fr	Baseline (single index)	.31	.48	.50
	Multiple Indexes	.29	.45	.47
	Concept + Single Index	.32	$.51^*$	.50
	ONB Dataset			
en	Baseline (single index)	.16	.26	.36
	Multiple Indexes	.15	.24	.35
	Concept + Single Index	$.17^*$	.27	.37
de	Baseline (single index)	.23	.35	.47
	Multiple Indexes	.23	.34	.49
	Concept + Single Index	$.24^*$	.35	.47
fr	Baseline (single index)	.15	.22	.31
	Multiple Indexes	.14	.20	.32
	Concept + Single Index	.15	.22	.31
	BNF Dataset			
en	Baseline (single index)	.25	.39	.45
	Multiple Indexes	.22	.34	.45
	Concept + Single Index	.25	.39	.45
de	Baseline (single index)	.24	.35	.45
	Multiple Indexes	.22	.32	.43
	Concept + Single Index	.24	.36	.45
fr	Baseline (single index)	.27	.38	.51
	Multiple Indexes	.25	.35	.50
	Concept + Single Index	.27	.37	.50

Table IV.14: Results of the three retrieval approaches that we submitted to the CLEF 2009 ad-hoc track: the baseline using a single index, retrieval based on multiple language-specific indexes and concept based retrieval combined with a single index. Statistical relevant improvements according to a paired t-test with confidence level .05 are marked with *.

Track	*Rank*	*Participant*	*Experiment DOI*	*MAP*
English	1st	chemnitz	CHEMNITZ.CUT_13_BILI_MERGED_DE2EN_9_10	40.46%
	2nd	hit	XTDD10T40	35.27%
	3rd	trinity	TCDDEENRUN3	35.05%
	4th	trinity-dcu	TCDDCUDEEN1	33.33%
	5th	karlsruhe	DE_INDEXBL	32.70%
French	1st	chemnitz	CUT_24_BILI_EN2FR_MERGED_LANG_SPEC_REF_CUT_17	25.57%
	2nd	karlsruhe	EN_INDEXBL	24.62%
	3rd	chesire	BIENFRT2FB	16.77%
	4th	trinity	TCDDEFRRUN2	16.33%
	5th	weimar	CLESA169283ENINFR	14.51%
German	1st	chemnitz	CUT_5_BILI_MERGED_EN2DE_1_2	25.83%
	2nd	trinity	TCDENDERUN3	19.35%
	3rd	karlsruhe	EN_INDEXBL	16.46%
	4th	weimar	COMBINEDFRINDE	15.75%
	5th	chesire	BIENDET2FBX	11.50%

Table IV.15: Official results of the bilingual TEL task at the CLEF 2009 ad-hoc track [Ferro and Peters, 2009].

CLEF 2009 Results. The results of our experiments are presented in Table IV.14. The results using multiple indexes show that this approach was not able to beat the baseline. Using a single index for the TEL records without language classification and topics only translated into the main language of each dataset achieved better performance compared to our approach based on indexes for each language and multiple translations of the topic to the matching languages.

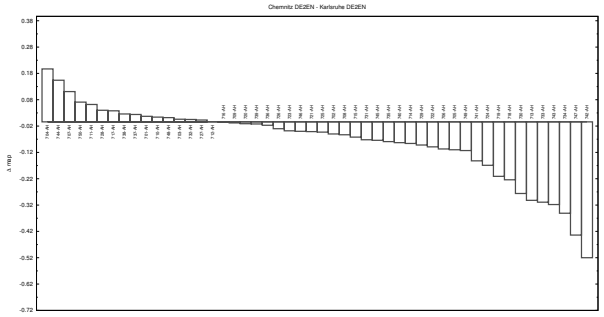
Another interesting result is that the combination of concept-based retrieval to the Machine Translation based retrieval was able to improve the retrieval in some cases. The improvement was significant according to a paired t-test with confidence level .05 for French topics on the BL dataset and English and German topics on the ONB dataset. However, in many cases the performance was similar to the baseline without statistical significance of the difference. We could therefore not reproduce the strong improvements for example presented in [Müller and Gurevych, 2008]. Müller and Gurevych [2008] used different datasets in their experiments with much longer documents compared to the TEL dataset. In the BL, ONB and BNF dataset many documents only consist of a few terms. This is a possible explanation of the different outcomes, as the CL-ESA retrieval model seems to have better performance on a search task with longer documents.

The official results of the bilingual TEL task at the CLEF 2009 ad-hoc track are presented in Table IV.15 [Ferro and Peters, 2009]. Our best runs achieved rank five on the English BL dataset, rank two on the French BNF dataset and rank three on the German ONB dataset. For the French BNF dataset, the differences in MAP of our best run compared to the run that is ranked first are not significant according to

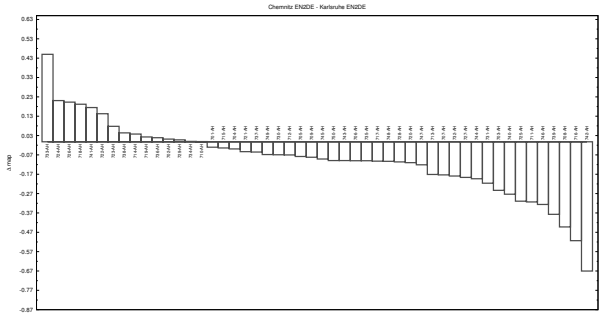
a paired t-test.

For a more detailed analysis, we compared the retrieval results of each topic. In Figure IV.16, the differences in Average Precision of our concept based retrieval approach and the best run that was submitted to the according track are plotted for each topic. The bars that are placed above the zero level represent topics for which our approach achieved better results than the best submitted run. Topics for which our approach failed to beat the best submitted run are represented by bars below the zero level.

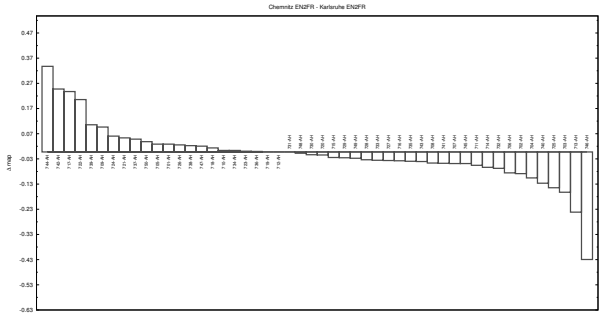
The plots in Figure IV.16 show that our concept-based retrieval approach is able to outperform the best run for several topics. However, for other topics the performance is worse. This allows the conclusion that concept-based retrieval has the potential to improve retrieval results. In future work, a further analysis is needed that studies the reasons of this fluctuation of performance. This includes the definition of characteristics of topics that are best supported by concept-based retrieval. Based on these characteristics, a classifier could then be used to predict the retrieval performance which allows to choose the most promising retrieval method for a specific topic.



(a) German topics on the English BL dataset.



(b) English topics on the German ONB dataset.



(c) English topics on the French BNF dataset.

Figure IV.16: Comparison of the *concept + single index* run to the best run of each bilingual TEL task at CLEF 2009. The differences in Average Precision are presented for each of the 50 topics that were used in this retrieval challenge.

Chapter V

Category-based Language Models for Multilingual Expert Retrieval

In Chapter IV, we presented Cross-lingual Explicit Semantic Analysis (CL-ESA), an approach that indexes text with respect to categories. This results in semantic representations of documents that can be used in concept-based retrieval systems. Thereby, the Wikipedia databases in several languages are used as data sources from which the concepts are extracted. These concepts define an external category system that we used for the CL-ESA approach. A different situation is given if the category system is defined in the context of the target dataset of the retrieval task. In contrast to an external category system defined for example by Wikipedia, the items of interest in the retrieval task are categorized and therefore define an internal category structure.

Examples of datasets with such an internal category structure are datasets from Social Question/Answer Sites (SQASs). In these community portals, questions and answers are usually categorized to a taxonomy which helps users to navigate and to find items in specific topic fields. This introduces an internal category structure on the questions and answers in the SQAS. The most prominent instance of a SQAS is Yahoo! Answers which will also be used in our experiments.

For retrieval scenarios defined on these datasets, an open question is how the internal category structure can be exploited to improve the retrieval performance. We propose different retrieval models that are based on categories in the dataset and therefore address this problem. These models are rooted in the theory of language models and combine different sources of evidence — with one of them being the categorical information. In our experiments we show that the new models outperform state-of-the-art retrieval approaches that are not aware of the category system. We therefore show the benefit of factoring background knowledge, in this case internal

category systems, into the retrieval process.

A further aspect we address in this chapter is the optimization of the combined retrieval models. We propose to use Machine Learning techniques to find optimal parameters to combine different sources of evidence.

New datasets such as datasets from SQASs allow to define new retrieval scenarios. In addition to the standard IR scenario of retrieving relevant text items given a query, other scenarios such as the retrieval of similar questions or the identification of answers of high quality are possible. In this chapter, we consider the problem of routing questions to experts. Given a new question, this problem is defined as identifying the expert who is most likely able to answer the question and who should therefore be contacted. We apply this problem to SQASs and consider all users of these portals as potential experts. The question routing problem is an instance of an IR problem as a relevance ranking of experts is needed for a new question. We refer to this retrieval problem of relevant experts given a specific question as *Expert Retrieval*.

In the following, we will first formally introduce the problem of Expert Retrieval. Then, we present a detailed definition of language models used for Information Retrieval. Thereby, we focus on Multilingual IR as well as Expert Retrieval. Afterwards, we define our approaches to combine different sources of evidence — mixture language models and Machine Learning based models. Finally, we present various experiments on a dataset extracted from Yahoo! Answers, a popular Social Question/Answer Site. The results show that our proposed retrieval models are able to benefit from the internal category system in this dataset for the task of retrieving relevant experts for new questions.

V.1 Expert Retrieval

In this chapter, we consider the task of Expert Retrieval (ER). In difference to the document retrieval scenario that was used in the preceding chapters, the information items to be retrieved are users of the system. Given an information need, relevant users have to be identified — which are then referred to as experts for this topic. The task of ER is defined as follows:

Definition V.1 (Expert Retrieval) *Expert Retrieval (ER) is a special case of Entity Search as defined in Definition I.9. In ER, the items of interest in the retrieval task are users. ER systems attempt to identify those people having the most* expertise *on a topic that is specified by a specific information need.*

Similar to document retrieval, the task of ER can be formalized as follows: given a set of experts E and an information need specified by query q, the expected retrieval result is a ranked list of experts $e_i \in E$ such that $rank_q(e_i) < rank_q(e_j)$ implies that expert e_i is more likely able to answer the information need than expert e_j.

To determine the expertise of users, we will use a text-centric definition of expertise in this thesis. We make the assumption that expertise can be captured reasonably enough by analyzing the content of the postings of experts. These postings are captured by the *text profiles* of experts:

Definition V.2 (Text Profiles) *Given expert e, the* text profile *of e is defined by the sum of all textual contributions or postings of e. This is also referred to as the* signature *of an expert e.*

In the context of a specific document collection D, the text profile of e is defined by the set of documents that are completely or partially written by e.

This definition of text profiles allows to apply established techniques from IR to the problem of ER. This will be elaborated in the following sections.

The text-centric definition of expertise is also motivated by the large amount of textual contributions of people to Web 2.0 sites. Many of these contributions can be used as evidence for expertise in specific topic fields, for example posted answers to questions in SQASs or modifications of Wikipedia articles. We thus characterize users of such portals by their textual contributions which allows to apply new retrieval scenarios such as ER to the Web 2.0.

ER has different applications in personal or corporate environments. In this thesis, we focus on applications in the context of the Web 2.0. Our main use case are SQASs. These portals are dedicated to answer complex information needs that might not be easily solved by existing IR systems, for example Internet search engines. In these portals, questions are answered by other users that are or claim to be experts in the specific topic field. Identifying potential experts for new questions can be used for *query routing*. The most relevant experts are then notified about the new questions in their area of expertise. This potentially leads to lower response times as relevant experts can act immediately on notification.

When considering international communities as for example in Yahoo! Answers, ER obviously calls for multilingual technologies. To support this statement we distinguish between *knowledge* and *contributions*. The knowledge of an expert about a specific topic is language-independent and can be shared to people of other languages. However, the contributions of experts about this topic are language-specific. Considering for example a query in language l_1, relevant experts with contributions in language l_2 will not be found. There is a language mismatch of the query to the text profiles of experts. Multilingual retrieval approaches offer solutions to this problem, as relevant experts for information needs and text profiles in arbitrary languages can be identified.

V.2 Language Models for IR

In Section II.2.3, we already briefly introduced language models and their application to IR. In the following, we will present more details and background of the

theory of language models. We also extend the basic models to support the ER task. This presentation of the theory of language models is inspired by the language model chapter of Manning et al. [2008] that gives a detailed introduction to this topic.

In this section, we will first present the basic theory of language models. We will further introduce different extensions that are relevant to the retrieval task considered in this chapter, namely Expert Retrieval on Social Question/Answer Sites. Then, we will present an historical overview of language models in IR. Finally, we propose new extensions that allow to define language models for multilingual retrieval and to exploit the semantics given by category systems.

V.2.1 Theory of Language Models

Basically, a language models is a probability distributions P over a vocabulary $V = \{t_1, \ldots, t_n\}$. As a property of probability distributions, the sum of probabilities of all terms has to be 1:

$$\sum_{t \in V} P(t) = 1$$

Interpreting a language model as a generative model, $P(t)$ defines the probability of drawing term t given this language model.

Language models can also be used to determine the probability of a sequence of terms. This is modeled as the subsequent drawing of terms from the probability distribution. In the context of IR, often a *unigram model* is assumed. In this model, term occurrences are independent of context. This simplifies the probability of a sequence of terms to the product of the term probabilities.

Example V.1 *Given a language model with probability distribution $P(t)$, the probability of the term sequence $t_1 t_2 \ldots t_k$ based on the unigram model is defined as:*

$$P(t_1 t_2 \ldots t_k) = P(t_1) P(t_2) \ldots P(t_k)$$

The estimation of language models is usually based on statistics derived from text corpora. In fields such as speech recognition, large corpora are used to estimate not only unigram models but also models that consider the context of terms, for example n-gram models. However, these context-sensitive models require large amounts of training data to overcome the data sparseness problem. In IR, language models are usually estimated on single documents. In most cases, this only allows to estimate unigram language models.

A common estimation of a language model $P(t|d)$ from document d is based on the maximum likelihood estimation of each term [Ponte and Croft, 1998]. This is based on the term frequency $\text{tf}_d(t)$ of term t in document d, normalized by the document length $|d|$ (see Chapter II):

$$P(t|d) = \frac{\text{tf}_d(t)}{|d|} \tag{V.1}$$

To apply language models to IR, a common approach is the *query likelihood model*. Given a query q, a retrieval system needs to identify the most relevant documents. In language models, this relevance is formalized by the conditional probability $P(d|q)$ of document d given query q. Using Bayes' law, this can be transformed to:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}$$

As $P(q)$ is constant for query q, this factor has no influence on the final ranking. In many retrieval systems, the a priori probability of documents $P(d)$ is assumed to be uniform, making this factor obsolete as well.

In analogy to the unigram model of documents, the query terms $t \in q$ are also assumed to be independent. Using the generative language model $P(t|d)$ of document d as defined in Equation V.1, the score of d in respect to query q is then defined as:

$$\text{score}_q(d) = P(d|q) \propto P(q|d) = \prod_{t \in q} P(t|d) \tag{V.2}$$

There are alternatives to the query likelihood model for using language models for IR. Another option is building both language models of documents and the query. The scores of document d is then computed by comparing the document model $P(t|d)$ to the query model $P(t|q)$. In Section IV.4, we already introduced KL-Divergence as distance measure of probability distributions, applied to concept-based retrieval. The KL-Divergence distance D_{KL} can also be used to define a term based retrieval model [Lafferty and Zhai, 2001]:

$$\text{score}_q(d) = -D_{\text{KL}}(q\|d) \propto -\sum_{t \in V} P(t|q) \log P(t|d)$$

The advantage of comparing document and query language models is the simple integration of more complex query models. Query models can be estimated not only using the actual query terms but also by integrating background knowledge. Often, similar techniques as the ones known from query expansion are used to build extended query models, for example Pseudo Relevance Feedback (see Definition II.2).

V.2.2 Smoothing

One problem of the query likelihood model is the conjunctive interpretation of queries. As the score of documents as defined in Equation V.2 is a product of the probabilities for each query term, any missing query term in document d will lead to a zero score of d. This is based on the estimation of the language model of d, which will assign zero probability to any term not occurring in d.

A common solution to this problem is *smoothing*. By extending the document language model to have non-zero probabilities for all terms in the vocabulary, missing query terms do not lead to zero scores in the retrieval process. The extended

document language models are usually based on statistical measures such as term distributions in the entire corpus.

An example for smoothing that is often applied in IR scenarios is *Jelinek-Mercer smoothing* [Jelinek and Mercer, 1980]. Using a background language model $P_{\text{bg}}(t)$, document language models are based on a mixture probability distribution:

$$P'(t|d) = (1-\alpha)P(t|d) + \alpha P_{\text{bg}}(t) \tag{V.3}$$

with α being the weight parameter for the background model. A maximum likelihood estimation on the corpus D can be used to estimate the background language model:

$$P_{\text{bg}}(t) = \frac{\sum_{d \in D} \text{tf}_d(t)}{\sum_{d \in D} |d|}$$

Another popular smoothing method is based on Dirichlet priors, which has also been successfully applied to IR scenarios [MacKay and Peto, 1995].

According to Manning et al. [2008], smoothing does not only solve the problems of the conjunctive interpretation of queries but also improves the overall performance of IR systems. They argue that smoothing adds important weights to terms that are similar to the inverse document frequency factors as used in vector space and probabilistic retrieval models.

V.2.3 Language Models for Expert Search

In this chapter, we consider the application scenario of expert search as defined in Section V.1. In contrast to standard IR settings, the items of interest in the retrieval task are defined by the set of experts E. These experts are described by a set of documents D that are related to E, for example by the authorship relation. As already defined in Definition V.2, the set of all documents related to expert e is called the text profile of e.

To apply language models to ER, the scoring function as defined in Equation V.2 has to be extended to support text profiles. In analogy to document retrieval, the score of expert e given query q based on language models is defined as:

$$\text{score}_q(e) = P(e|q) \propto P(q|e) = \prod_{t \in q} P(t|e)$$

In order to estimate the conditional probability $P(t|e)$ of term t given expert e, we use the language models of the documents in D:

$$P(t|e) = \sum_{d \in D} P(t|d)P(d|e) \tag{V.4}$$

with $P(d|e)$ modeling the relation of d and e. In general, it is not possible to estimate $P(d|e)$ directly as the language model of e is not known. Therefore, we apply Bayes'

Theorem in order to estimate the probability $P(d|e)$:

$$P(d|e) = \frac{P(e|d)P(d)}{P(e)}$$

The priors $P(d)$ and $P(e)$ are often assumed to be uniformly distributed, i.e. $P(d) = \frac{1}{|D|}$ and $P(e) = \frac{1}{|E|}$. In the case that each document has exactly one author out of E, the probability $P(e|d)$ of expert e given document d can be modeled as:

$$P(e|d) = \begin{cases} 1 & \text{if } e \text{ is author of } d \\ 0 & \text{else} \end{cases}$$

In other cases with several authors per document, more complex estimations are required. Overall, the scoring function based on these simplifying assumptions is defined as:

$$P(q|e) \approx \prod_{t \in q} \sum_{d \in D} P(t|d)P(e|d)$$

Combined with smoothing, this model has been proposed by Balog et al. [2009a]. Their *Model 1* is defined as:

$$P(q|e) \approx \prod_{t \in q} \left[(1-\alpha) \left(\sum_{d \in D} P(t|d)P(d|e) \right) + \alpha P_{\text{bg}}(t) \right]$$

with α being the smoothing weight as presented in Equation V.3. The background language model $P_{\text{bg}}(t)$ is again estimated on the entire corpus D. We will refer to this model as P_{LM}.

V.2.4 Historical Overview

The theory of language models applied to IR has been introduced by Ponte and Croft [1998]. They proposed to use Maximum Likelihood Estimation to model the document language models as presented in Equation V.1. They also suggested to use smoothing based on corpus statistics as defined in Equation V.3.

The research in ER was mainly driven by the enterprise track at TREC (see for example [Craswell et al., 2005]). In this context, different entity retrieval systems were published that are based on language models.

Cao et al. [2005] proposed a two-stage language model. They use the notion of two-stage as they model expertise by documents that are related to experts. This is similar to the estimation of the expert language model as presented in Equation V.4.

Balog et al. [2009a] defined two different language models for ER. Model 1 or *the expert language models* are based on textual profiles of experts to estimate the language models of experts as in our approach. We thus follow Model 1 and propose several extensions thereof. Model 2 or *the document language models* are based on the retrieval of relevant documents that are then used to identify associated experts.

Balog et al. compare both systems using different extensions on several datasets. Their results show that no clear preferred model can be identified as the differences of performance depend on the extensions and on the used dataset.

Petkova and Croft [2006] suggest a hierarchical language model for ER. In principle, they use Model 1 as defined by Balog et al. [2009a]. In addition, they propose to use evidence of different document collections to build a combined model of expertise. These different collections might also be defined by subsets of the original collection. Their combined model is based on a linear combination of language models that are each based on a single sub-collection. The sub-collections as suggested by Petkova and Croft are thereby defined by the categories. In Section V.4, we also present a combining approach that uses linear combination to define a mixture model of probabilities.

V.3 Extensions of Language Models

In this chapter, we consider the scenario of multilingual ER. Therefore, we propose an extension of the language model retrieval framework that supports multilingual documents and queries. The dataset we use in our experiments, which is extracted from Yahoo! Answers, has an internal category structure. In order to use this information in the retrieval process, we further define language models for category systems that allow to exploit the category structure in the dataset.

V.3.1 Language Models for MLIR

To apply language models to multilingual retrieval scenarios, we propose to use query translation as introduced in Chapter II. Given the set of document languages $L = \{l_1, \ldots, l_n\}$, query q in language l_q is translated to each language l_i using a Machine Translation system. The combined query q^* is then defined as the union of terms of all the translated queries:

$$q^* = q \cup q_{l_q \to l_1} \cup q_{l_q \to l_2} \cup \ldots$$

In this translated query, the language $\text{LANG}(t)$ of each query term t is determined by the language of the translation that introduced the term.

The translated query q^* has also implications on the estimation of document and background language models. The maximum likelihood estimation of the document language model as presented in Equation V.1 is defined as follows for translated query terms:

$$P(t|d) = \begin{cases} \frac{\text{tf}_d(t)}{|d|} & \text{if LANG}(t) = \text{LANG}(d) \\ 0 & \text{else} \end{cases}$$

This means that query terms of other language as the document language are generated with zero probability.

The background language model — used for example for smoothing — is also adapted to the multilingual query. In essence, different language models are built for each language. For example for language l, only the subset of documents in language l are considered: $D_l = \{d \in D \mid \text{LANG}(d) = l\}$. The background language model for query terms in language l is estimated using D_l. We refer to this language specific background model as $P^*_{\text{bg}}(t)$, which is based on different language models for each language.

V.3.2 Language Models for Category Systems

As part of our ER scenario introduced in Section V.1, we assume that the documents, which are defined by the contributions of experts and are therefore part of the text profiles of experts, are organized in a categorical structure. In Chapter I, we motivated such category systems as potential knowledge source for retrieval systems. Our suggested approach to integrate this knowledge is based on the extension of language models using generative models of categories.

In the set of categories C, each *category* $c \in C$ is a subset of corpus D: $c \subseteq D$. Categories are used to assign documents to specific topic fields. Thereby, categories do not have to be disjoint. They may contain common documents or even include other categories, resulting in a hierarchical structure of categories.

We define language models on categories based on the textual evidence of the documents in each category. Using Maximum Likelihood Estimation, the language model $P(t|c)$ of c is then defined as:

$$P(t|c) = \frac{\sum_{d \in c} \text{tf}_d(t)}{\sum_{d \in c} |d|} \tag{V.5}$$

This generative model defines the probability of category c generating term t.

To integrate language models of categories in the ER model, we suggest an approach similar to summing over documents as presented in Equation V.4. Assuming independent query terms, the retrieval model mainly depends on the estimation of $P(t|e)$. We propose to sum evidence for all categories given expert e to estimate $P(t|e)$:

$$P(t|e) = \sum_{c \in C} P(t|c)P(c|e) \tag{V.6}$$

Intuitively, $P(t|c)$ models the probability that query term t is generated given the language model of category c. This probability is weighted using the conditional probability $P(c|e)$ of category c given expert e. Thus, the language models of categories with higher probability given expert e have more influence in the generative model $P(t|e)$.

Example V.2 *We consider the two extreme cases of expert e_1 and e_2. e_1 is only related to category c_1, i.e. $P(c_1|e_1) = 1$ and $P(c_i|e_1) = 0$ for $c_i \neq c_1$. For*

expert e_2, all categories have the same probability $P(c_i|e_2) = \frac{1}{|C|}$. Then, $P(t|e_1)$ corresponds to the language model of c_1:

$$P(t|e_1) = P(t|c_1)$$

and $P(t|e_2)$ is the average of all category language models:

$$P(t|e_2) = \frac{1}{|C|} \sum_{c \in C} P(t|c)$$

$P(t|c)$ is estimated via the term frequency of t in all documents d contained in category c as presented in Equation V.5. These language models directly exploit the category structure of corpus D.

The remaining problem is the estimation of $P(c|e)$ — the probability of category c given expert e. In this chapter, we propose different approaches to estimate $P(c|e)$. These approaches are specific to the application scenario as they exploit further background knowledge. In our retrieval models, we use measures that quantify the popularity of an expert within a given category. In detail, we define two different popularity models:

Frequency based Popularity Model. Given a category $c \subseteq D$ containing $|c|$ documents, each expert is associated to a subset of these documents, for example the documents authored by this expert. For expert e, we define this set of associated documents in category c as c_e. Using the size of c_e, the frequency based popularity model is defined as:

Definition V.3 (Frequency Popularity Model (PM_{freq})) *The popularity of expert e in category c is defined by the size of the set of documents c_e associated to e:*

$$PM_{freq}(e, c) = \frac{|c_e|}{|c|}$$

Expert Network based Popularity Model. In many application scenarios — especially in the Web 2.0 context — experts are connected to each other. These connections are either explicit, for example personal contact lists or friendship relations, or implicitly derived from the data. An example for an implicit relation is co-authorship. Documents having multiple authors define a relation between these authors.

In this chapter, our experiments are based on Social Question/Answer Sites. In this context, users post questions that are answered by other users. This defines implicit relations between questioners and answerers. More details will be presented in Section V.5.

Using arbitrary connections between experts, we define the *expert network* as follows:

Definition V.4 (Expert Network) *The set of experts E and the relation $\Gamma \subset E \times E$, which formalizes the connections between experts, define the expert network $G = (E, \Gamma)$. Nodes are defined by the set of experts E and edges are based on Γ. Two experts e_1 and e_2 are linked iff $(e_1, e_2) \in \Gamma$.*

In category systems and using implicit expert connections based on documents, the category-specific expert network G_c for category c is only based on connections that are defined by the documents in c.

Expert networks allow to apply measures of centrality known from graph theory. The goal is to exploit these measures to quantify the popularity, either within the subgraph for one category or for the entire graph. In this section, we propose to use the PageRank algorithm [Brin and Page, 1998] as instance of an approach to compute the centrality of nodes. We chose PageRank as the PageRank algorithm computes a single value of centrality for each node which is needed to define the following popularity model:

Definition V.5 (PageRank Popularity Model ($PM_{pagerank}$)) *Applied to the expert network $G = (E, \Gamma)$, the PageRank algorithm computes a centrality value $PAGERANK(G, e)$ for each expert e. The PageRank popularity model can also be applied to category-specific expert networks G_c. We define the PageRank popularity model based on these centrality values:*

$$PM_{pagerank}(e, c) = \frac{PAGERANK(G_c, e)}{\sum_{e' \in E} PAGERANK(G_c, e')}$$

Informed vs. Non-informed Search In category systems, documents are classified to categories. This also allows the a priori classification of topics to a specific category in the retrieval scenario. While statistical classifiers can be used to classify topics, the manual selection of the topic category is also an option in many application scenarios. The user formulating a query is then required to also specify the most appropriate target category. This is for example often implemented in SQASs.

The information about the topic category is valuable in the search process. The retrieval results can be improved by only considering documents and experts in the target category, which will be shown in our experiments.

In the following, we distinguish between *informed* and *non-informed* approaches. Informed approaches assume that the target category of a topic is known in advance. In the retrieval model, these approaches also exploit this information. Non-informed approaches do not depend on the category classification of topics.

V.4 Combining Sources of Evidence

The expert retrieval models presented in Section V.2 are all document centric models. The only evidence for expertise is given by the set of documents and their relation

to the set of experts. However, in many application scenarios, many more features are available that can be used to define expertise. An example is the scenario of ER in SQASs as proposed in this chapter. The complex relations between experts, ratings of questions or answers and other features of these portals could be used to identify relevant experts with respect to specific topics. An open challenge is how these features can be integrated into retrieval models and how the evidence provided by them can be combined.

In the following, we propose a principled way of combining different sources of evidence in the retrieval framework of language models. This allows to use different estimations of expertise in the overall scoring function.

The combination of different sources of evidence requires a weighting of the different parameters. This calls for the application of Machine Learning techniques to identify optimal parameter settings. We present different approaches to define retrieval functions using supervised ML.

In this section, we first present a mixture language model that allows to combine different estimations of expertise for given query terms. Then, we propose a discriminative model that reduces the ranking problem to a regression problem, which can be solved using ML techniques. Finally, we present an alternative approach based on the principles of Learning to Rank. In this approach, a classifier is used to compute preferences between pairs of experts, which are then used to define a scoring function for the ER task.

V.4.1 Mixture Language Models

To integrate different sources of evidence for ER, the main idea is to use different estimations for the probability distribution $P(t|e)$. For each source i, $P_i(t|e)$ defines the conditional probability of query term t given expert e based on the information given by this source of evidence.

In order to integrate the different estimations $P_i(t|e)$, we propose a mixture language model (MLM) with weight parameters α_i as for example also used in [Petkova and Croft, 2006]:

$$P(t|e) = \sum_i \alpha_i P_i(t|e)$$

with $\sum_i \alpha_i = 1$. This defines a valid conditional probability distribution. The α_i weights determine the influence of each component or source of evidence in the mixture model.

In the ER scenario, this mixture model is used to define the scoring function. One source of evidence is defined by the smoothing factor $P_{\text{bg}}(t)$, which is based on the background language model. The weight of the background model is determined by the weights of the other models, resulting in the following scoring function that extends Model 1 of Balog et al. [2009a]:

$$P(q|e) = \prod_{t \in q} \left[\sum_i \alpha_i P_i(t|e) + (1 - \sum_i \alpha_i) P_{\text{bg}}(t) \right]$$

with $0 < \alpha_i < 1$ and $\sum_i \alpha_i < 1$.

The novelty of this approach lies in the different models used to estimate $P_i(t|e)$ based on different sources of evidence. In the following, we present different models in the context of expert search which will also be evaluated in our experiments.

Text Profile Model. This is a generative model that quantifies the probability of an expert e generating a query term t on the basis of its text profile, consisting of all documents that are associated to this expert:

$$P_{\text{profile}}(t|e) = \sum_{d \in D} P(t|d)P(d|e)$$

This source of evidence corresponds to the document-based expert search model, which is for example presented in [Balog et al., 2009a].

Category-restricted Text Profile Model. Given a category system in the dataset that classifies all documents to a set of categories C, this model can be used for informed search scenarios. In this case, the target category c of query q is known in advance. Based on this information, the set of documents can be restricted to documents in this category, which results in the following estimation:

$$P_{\text{profile+c}}(t|e) = \sum_{d \in c} P(t|d)P(d|e)$$

This model is based on category-specific text profiles of experts. Only documents in the target category are considered in order to estimate the relevance of experts. Intuitively, the language model of expert e is based on the language models of all documents d_i that are related to e. By only using documents in the target category c, the documents in other categories have no influence on the language model of e. This is potentially helpful in the retrieval task as these documents — which are not classified to the target category — might introduce noise to the language models of experts in respect to the given query.

Category Model. As presented in Section V.3, language models of categories can be defined using the text of all documents in each category. These language models can then be used for an implicit classification of query terms, resulting in the following category-based estimation of $P(t|e)$ as defined in Equation V.6:

$$P_{\text{cat}}(t|e) = \sum_{c \in C} P(t|c)P(c|e)$$

The probability $P(c|e)$ of category c given expert e is estimated by popularity models as presented above: $P(c|e) \approx \text{PM}(e, c)$, which are either based on the share of documents in category c associated to e (PM_{freq}) or the PageRank scores of expert e in category c ($\text{PM}_{\text{pagerank}}$, see Section V.3.2).

Informed Category Model. In an informed scenario — where the category c of the query is known — the category language model $P(t|c)$ is obsolete. This function will be uniform for the target category and 0 otherwise. Using again the popularity models as in the category model, this results in the following estimation of $P(t|e)$:

$$P_{\text{cat+c}}(t|e) = P(c|e) \approx \text{PM}(e, c)$$

V.4.2 Discriminative Models

Using the MLM presented above, finding optimal values of the weight vector is essential. When using several sources of evidence, exploring the parameter space using heuristic methods is time consuming and potentially leads to local maxima. This calls for the use of optimization techniques known from Machine Learning to combine several sources of evidence. In summary, we propose to reduce the retrieval task to a regression problem.

Machine Learning (ML) algorithms expect *feature representations* of instances. In the ER task, feature representation of experts depend on the topic and are therefore defined on expert-topic tuples:

Definition V.6 (Feature Representation of Experts) *The feature representation of expert-topic tuples in a real-valued feature space $\mathbb{R}^n$ is defined by the mapping function*

$$F(e, q) : E \times V^k \rightarrow \mathbb{R}^n$$

with E being the set of experts and V the vocabulary that is used to formulate any query q.

When combining different sources of evidence, each source i defines a mapping function $F_i(e, q)$ that defines the source-specific features. The combined feature representation is then defined by the concatenated feature vector:

$$F(e, q) = (F_1(e, q)^T, F_2(e, q)^T, \dots)^T$$

Intuitively, each of the features defined in $F(e, q)$ measures the relevance between expert e and query q. This allows to integrate different measures of relevance, which are for example based on different relevance models or on different sources of evidence.

Using the feature representation of experts given a specific topic that is represented by query q, the ranking of experts can be reduced to the following regression problem:

Definition V.7 (ER as Regression Problem) *The problem of ranking experts E for a given query q can be reduced to a regression problem of learning the optimal parameters for a function expertise $: \mathbb{R}^n \rightarrow [0, 1]$. The input is thereby given by*

Feature	*Description*
MIN	Minimal probability: $\min_{t \in q} P_i(t\|e)$
MAX	Maximal probability: $\max_{t \in q} P_i(t\|e)$
AVG	Average probability: $\frac{\sum_{t \in q} P_i(t\|q)}{\|q\|}$
MEDIAN	Median probability in the sorted values of $\{P_i(t\|e) \mid t \in q\}$
STD	Standard deviation in the set $\{P_i(t\|e) \mid t \in q\}$

Table V.1: Aggregated features that are defined by the conditional probability distribution $P_i(t|e)$. These features describe properties of expert e given query $q = (t_1, t_2, \dots)$ according to generative model i.

the feature representation $F(e, q)$ of each expert e. The experts can then be ranked according to expertise$(F(e, q))$, i.e.

$$\textit{rank}(e_1) \leq_q \textit{rank}(e_2) \leftrightarrow \textit{expertise}(F(e_1, q)) \geq \textit{expertise}(F(e_2, q))$$

The training data for such regression problems is given by topics with existing relevance assessments. Assuming a binary relevance function, the set of relevant experts E_q^{rel} and non-relevant experts $E_q^{\text{non-rel}}$ for query q are used as training examples. The expected outcome of the classifier function is then defined as

$$\text{expertise}(F(e, q)) = \begin{cases} 1 & e \in E_q^{\text{rel}} \\ 0 & e \in E_q^{\text{non-rel}} \end{cases}$$

An important success factor for the application of ML algorithms is the engineering and selection of appropriate features. In the following, we present the features that are used in our experiments. Afterwards, we describe the different regression and classification models that we apply to the problem of ER.

Feature Design. As explained above, feature representations of candidate experts are used as input for the regression function. Each feature of the vector $F(e, q)$ (see Definition V.6) describes a property of expert e given the specific query q. Features are defined on pairs of experts and queries, which was already proposed by Joachims [2002] in his Learning to Rank approach.

Additionally, the set of features that is used to classify experts has to be identical for different queries. This implies that features have to be independent on the number of query terms. The reason for this is that we want the classifier to work for a query with an arbitrary number of terms. Therefore the features we propose are aggregated values for all query terms.

Given expert e and query q, we use the following features to define $F(e, q)$:

Language Model Features: As described above, different sources of evidence are used to model the probability of query term t being generated by expert e:

Evidence	*Features*	*Count*
Language Models	$[P_{\text{profile}}]$, $[P_{\text{profile+c}}]$, $[P_{\text{cat}}]$, $[P_{\text{cat+c}}]$, $[P_{\text{bg}}]$	25
Products of LMs	$[P_{\text{profile}} \times P_{\text{profile+c}}]$, $[P_{\text{profile}} \times P_{\text{cat}}]$, $[P_{\text{profile}} \times P_{\text{cat+c}}]$, ...	50
	$[P_{\text{profile}} \times P_{\text{profile+c}} \times P_{\text{cat}}]$, $[P_{\text{profile}} \times P_{\text{profile+c}} \times P_{\text{cat+c}}]$, ...	50
Prob. Models	BM25, TF.IDF, DLH13 (on expert text profiles)	3
total		128

Table V.2: Summary of all features used in the discriminative model. For language models, five aggregated features where generated for the conditional probability distribution $P_i(t|e)$ over query terms: MIN, MAX, AVG, MEDIAN and STD.

$P_i(t|e)$. We use this probability distribution to define aggregated features for all query terms. These aggregated features (namely *minimum*, *maximum*, *average*, *median* and *standard deviation*) are described in detail in Table V.1. We thus assume the generation of any query term to be conditionally independent from the generation of other query terms (see [Gao et al., 2005] for an approach modeling such dependencies).

Products of Language Model Features: As the above aggregate features do not capture the dependencies between different sources of evidence on term level, we additionally consider products of these conditional probability distributions. For example given two evidence sources i and j, we define the combined distribution as $P_{ij}(t|e) = P_i(t|e)P_j(t|e)$. The aggregated features as described in Table V.1 are then computed based on P_{ij}. In the experiments in Section V.5, we add features for all permutations of up to three evidence sources. The dependencies on term level of the different language models are therefore captured in the feature representation of experts.

Probabilistic Model Features: In addition to the aggregated features over query terms for the different generative models, we also use standard IR retrieval scores based on the text profile of each expert. Features are then defined as the score of expert e given query q. We used in particular retrieval models based on BM25, TF.IDF and DLH13 term weighting in combination with Z-Score normalization for multilingual retrieval (see Chapter II for more details of these retrieval models), resulting in exactly one feature for each retrieval model.

All features uses for the discriminative model are summarized in Table V.2. In analogy to the mixture language model, we use two different generative models as sources of evidence for expertise: a generative model based on the text profiles of experts (P_{profile}) and a category generative model (P_{cat}). In the informed scenario, we also assume that the topic category is known and also use text profiles restricted to the target category ($P_{\text{profile+c}}$) and the category model of the target category ($P_{\text{cat+c}}$). As described above, the term dependencies across the different generative models

are modeled using products of query term probabilities for all permutations of up to three evidence sources. Finally, we also add features based on probabilistic retrieval models.

Classifiers. The discriminative model is implemented on the basis of a regression model. The output of regression models are continuous real values that can be directly used as scores of experts. In contrast to regression, the output of standard discrete or binary classifiers is not qualified to be used as scores. As all relevant experts are mapped to a single score value, they can not be further distinguished for ranking. As we show in our experiments, this usually leads to poor retrieval results.

However, discrete or binary classifiers allow to determine the probability of instances belonging to the positive or relevant class. This probability value can be used to define the score of experts e for query q:

$$\text{score}_q(e) = P(X = \text{relevant}|q, e)$$

While this makes any classifier applicable in principle to the expert retrieval task, our experiments have shown that discrete classifier functions performed much worse than regression-based classifiers.

In order to rank experts, we used the following regression models and classifiers:

Multilayer Perceptron (MLP): MLPs are instances of regression functions. They are based on different layers of connected *perceptrons*. Each perceptron has inputs from the preceding layer and computes its output value using a sigmoid function. In our experiments, we used a MLP layout with one hidden layer. The number of perceptrons k in the hidden layer is determined by the number of features n we used to represent experts: $k = (n + 2)/2$.

Logistic Regression: As a further regression function, we used the multinomial logistic regression model with a ridge estimator proposed by le Cessie and van Houwelingen [1992].

J48 Decision Tree: The J48 decision tree is a discrete classifier, which is a pruned version of the C4.5 decision tree [Quinlan, 1993]. We use this classifier to compare the performance of regression functions to discrete classifiers in our experiments.

V.4.3 Learning to Rank

An alternative approach to the discriminative model for using ML in IR was proposed by Joachims [2002] as *Learning to Rank*. Instead of directly classifying experts given the query q, the classifier is used to determine the *preference* of two experts in the context of q. The input feature vectors of this classification model therefore represent the difference of two experts. The learning to rank approach is defined as follows:

Definition V.8 (Learning to Rank) *Given a question q, the Learning to Rank approach compares pairs of experts to compute a preference of which expert to rank before the other. The ranking of experts can then be reduced to the following preference function:*

$$\mathit{pref} : E \times E \times Q \rightarrow \{0, 1\}$$

with pref$(e_1, e_2, q) = 1$ meaning that e_1 should be ranked before e_2 for query q. The preference function can be implemented by a classifier. This allows to use supervised ML to optimize this function based on training data.

The work of Joachims [2002] is motivated by the task of ranking documents where click-through data is used as relevance assessments in order to derive training pairs. Click-through data is based on the decision of users to click on one or several of the presented documents. As the users are presented a ranked list of top documents, these decisions can be interpreted as preferences of the clicked documents w.r.t. the documents presented before these in the ranked list. Learning to Rank allows to design a retrieval model that are optimized using training pairs of documents that are derived from this click-through data.

As classifier, Joachims used *RankSVM* — an implementation of a support vector machine that is adapted to the learning to rank problem. While training is performed on pairs of experts with the preference information, the actual scores in the retrieval process are computed for single experts. For our learning to rank retrieval model, we use a different approach. We refer to standard ML models that are also trained on pairs of experts. For retrieval however, we evaluate again pairs of experts to compute the scores that determine the ranking of experts. This will be explained below in more detail.

Similar to the discriminative model, feature design is also an important aspect of the learning to rank approach. We use the same features as presented for the discriminative model, which are adapted to support pairs of experts. The exact definition of the feature vector will be presented in the following. Afterwards, we describe our approach to define ranking of experts based on the output of the learning to rank classifier. Finally, we list the ML models that were used to implement this retrieval approach.

Feature Design. For the Learning to Rank approach, features of pairs of experts given a query are needed as input to the preference function. We will use the same features as defined above in the discriminative approach to model feature vectors of experts given a query. For expert e_1 and e_2 and query q, the learning to rank features vector $F'(e_1, e_2, q)$ is then defined as the vector difference of the features vectors $F(e_1, q)$ and $F(e_2, q)$:

$$F'(e_1, e_2, q) := F(e_1, q) - F(e_2, q)$$

Ranking. The learning to rank classifier determines the preference of two experts. Using the pair-wise comparison of all experts, we propose to use a *voting system*

to define scores of experts. This voting system defines the score of expert e as the number of experts that should be ranked lower than e based on pref:

$$\text{score}(e, q) = |\{e' \in E \mid \text{pref}(e, e', q) = 1\}|$$

As the pairwise comparison of all experts is not scalable for large sets of experts, we apply the learning to rank approach to re-rank a set of candidate experts. Using an alternative retrieval model, first a fixed number of experts is selected. In a second step, these experts are than ranked according to the preference function which results in the final ranking.

Classifiers. As the preference function used in the learning to rank approach is a binary decision function, no continuous output of the classifier is needed. The output of binary classifiers — such as the J48 decision tree — can directly be used as the preference function.

We also use the regression functions as used for the discriminative approach for the learning to rank approach, namely MLPs and Logistic Regression. This allows for the comparison of the different retrieval approaches and feature representations without changing the underlying ML model. Using the output of the regression functions, the binary preference function is defined using a fixed threshold for positive and negative decisions.

In contrast to Joachims [2002], we do not use a special classifier as RankSVM. Our focus is clearly on feature design and on the different retrieval models. We analyze the retrieval performance using different sets of features, based on single experts and on differences between experts. We therefore assume that standard classifiers are sufficient for these experiments.

V.5 Experiments

The experiments to evaluate the different retrieval approaches presented in Section V.4 are based on an expert retrieval scenario as presented in Definition V.1. We use a dataset that is derived from a crawl of Yahoo! Answers — an instance of a Social Question/Answer Site (see Definition I.4). This dataset consists of questions, answers to these questions, a categorization of questions and references to the authors of both questions and answers.

In the context of the Conference on Multilingual and Multimodal Information Access Evaluation (CLEF) 2010, we organized the Cross-lingual Expert Search (CriES) retrieval challenge.[1] Retrieval challenges define a specific retrieval task with standardized experimental settings. This includes a dataset and a set of topics with according relevance assessments (see Section II.5). For the CriES challenge, we used the crawl from Yahoo! Answers to build a dataset and selected a set of topics

[1] `http://clef2010.org/`, `http://www.multipla-project.org/cries` (last accessed April 8, 2011)

that are suitable for the specific retrieval task of multilingual Expert Retrieval. These settings are also used for the experiments in this chapter. In our evaluation, we will use the ground truth that is based on relevance assessments that were created in the context of CriES.

We define several retrieval baselines that our proposed approaches are compared to. This includes basic retrieval models but also the results of the other groups that participated at the CriES challenge.

In summary, the main objectives of our experiments are the following:

1. Analysis of the differences of the informed vs. the non-informed scenario. Does the information about the topic category help to improve the retrieval performance?

2. Presentation of the retrieval results using the mixture model in comparison to the baseline approaches. These baseline approaches include standard MLIR models and the retrieval systems used by participating groups at the CriES pilot challenge.

3. Presentation of the retrieval results using the discriminative model. This also includes the analysis of the training process and the resulting learned ML model. The results of the discriminative model are compared to the mixture model and the baseline approaches.

4. Presentation of the retrieval results using the Learning to Rank model. These results are compared to the results of the discriminative model that is based on a similar feature representation of experts.

V.5.1 Yahoo! Answers

Yahoo! Answers[2] is an instance of a Social Question/Answer Site (see Definition I.4). In summary, users of the portal post questions that are answered by other users. The detailed process of posting questions and answers as well as the rating and rewarding system will be described in the following.

Size and Market Share. According to Yahoo! Answers staff, the portal reached 200 million users in December 2009.[3] They also claim that 15 million users are visiting the portal each day.[4] The high number of visitors is confirmed by other sources. According to Google ad planner for example, the portal attracts 1.1B page views per month.[5]

[2] `http://answers.yahoo.com/` (last accessed April 8, 2011)

[3] `http://yanswersblog.com/index.php/archives/2009/12/14/yahoo-answers-hits-200-million-visitors-worldwide/` (last accessed April 8, 2011)

[4] `http://yanswersblog.com/index.php/archives/2009/10/05/did-you-know/` (last accessed April 8, 2011)

[5] Statistics from 2010/05/17

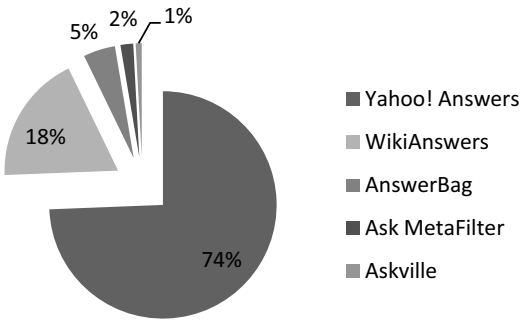


Figure V.1: Market share of Yahoo! Answers in the market of Social Question/Answer Site, measured by the number of visits in the US market during one week. Source: Hitwise USA, March 2008.

The market share of different SQASs sites is presented in Figure V.1. According to Hitwise USA, Yahoo! Answers has a market share of 74% in this segment.[6] The popularity and leading position of Yahoo! Answers clearly motivates to use the data in this portal for research. Yahoo! published an official crawl of Yahoo! Answers for research purposes. We used this dataset to define a corpus for multilingual ER, which will be described in Section V.5.2.

Features of the Yahoo! Answers Site. Yahoo! Answers has a defined process of posting and answering questions. A new question is *open* for a specified time period — initially for three days which can be extended up to eight days. In this period, users are allowed to post answers to this question. During this time, the questioner is able to pick a *best answer* out of the submitted answers. Selecting a best answer *closes* the questions, *i.e.* no further answers can be submitted.

The rating system implemented in Yahoo! Answers allows to rate questions and answers. Questions can be marked with stars — indicating interesting questions. Answers are rated using *thumbs up* or *thumbs down* tags. In the case that the questioner does not select a best answer, the number of received thumbs is used to automatically determine the best answer. The ratings of answers can be seen as voting for an answer to get the best answer. Additionally, the portal allows to post comments to answers.

The status of users in Yahoo! Answers is based on a rewarding system. Initially, the number of questions and answers that can be submitted to the portal by each user is limited. By gaining points that raise the status level of a user, these personal limits

[6] `http://www.marketingcharts.com/interactive/question-answer-site-visits-up-118-yahoo-answers-is-clear-leader-3891/` (last accessed April 8, 2011)

are extended. Points are earned for example if a submitted answer is selected as best answer. The level of users is also expressed by visual changes in the avatars of users, for example by an orange badge.

The rewarding system is designed to motivate users to submit questions and answers of high quality. Yahoo! Answers is a social community portal in which users have a social status. Increasing the level of a user can therefore be seen as a raise of her social status in this community. This is an important incentive for users to contribute and is probably also a key factor of the success of Yahoo! Answers.

Quality of Answers. Harper et al. [2008] analyzed the quality of answers in SQASs. They apply different measures to estimate the quality of answers, for example number of answers and the length of answers. Additionally, human assessors judged the quality of answers and the effort that was probably made to author the answers. In their study, they compared SQASs to non-free question answering portals and to portals that directly route questions to matching experts. Non-free question answering portals differ to SQASs as questioners bid money on their questions that is received by the answerer. The main outcomes of Harper et al. [2008] according to the quality of answers are:

- The quality of paid answers in non-free question answering portals is better compared to SQASs in the case that enough money is bid on questions. If the amount is too low, the answers given in SQASs have superior quality. Overall, the diversity of answers is higher in SQASs as more answers per questions are posted.
- The quality of answers in question answer portals — SQASs and non-free portals — is better compared to the answers in portals that route questions to single experts. In these portals, the percentage of questions that are not answered at all is also higher. The aspect that is not covered in the analysis of Harper et al. is the selection of appropriate experts for given questions. The ER approach suggested in this thesis aims at identifying experts that will most probably give high quality answers to new questions.

Knowledge Partner Program. In Yahoo! Answers, an effort to raise the quality of answers is made through the *knowledge partner program*. Any company or organization can offer professional-quality knowledge by answering questions in a specialized field. In return, they are allowed to mention their own products or services and visual features such as links or logos are displayed in the context of their answers.

V.5.2 Dataset

The dataset that we use to evaluate our ER models is is a subset of an official crawl of Yahoo! Answers, namely the *Yahoo! Answers Webscope dataset* that was introduced

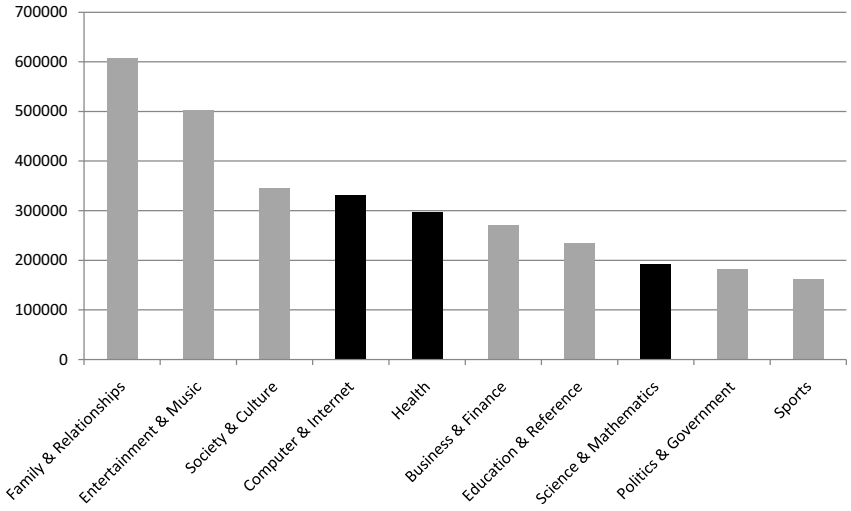


Figure V.2: Number of questions in the largest categories in the Yahoo! Answers Webscope dataset.

by Surdeanu et al. [2008].[7] This crawl contains 4.5M questions with 35.9M answers. For each question, one answer is marked as *best answer* (the selection process of best answers is described in Section V.5.1). The dataset contains ids of authors of questions and best answers, whereas authors of non-best answers are anonymous. Questions are organized into categories which form a category taxonomy.

In the context of the CriES workshop, we used this dataset to create a corpus that can be used to evaluate multilingual ER systems. We organized an open IR challenge, the CriES pilot challenge, that allows the comparison of retrieval system from different participating groups [Sorg et al., 2010].

Description. The dataset used in the CriES pilot challenge is a subset of the Yahoo! Answers Webscope dataset, considering only questions and answers in the topic fields defined by the following three top level categories including their sub categories: `Computer & Internet`, `Health` and `Science & Mathematics`. An overview of the largest categories in this dataset is given in Figure V.2. As our approach is targeted at the ER problem, we chose very *technical categories* yielding a dataset with a high number of technical questions. These questions require domain expertise to be answered. As many questions in the dataset serve the only purpose of

[7] This dataset is provided by the Yahoo! Research Webscope program (see `http://research.yahoo.com/` (last accessed April 8, 2011)) under the following ID: *L6. Yahoo! Answers Comprehensive Questions and Answers (version 1.0)*

Category	*Questions*	Language share *EN*	*DE*	*FR*	*ES*
Comp. & Internet	317,074	89%	1%	3%	6%
Health	294,944	95%	1%	2%	2%
Science & Math.	185,994	91%	1%	2%	6%

Table V.3: The share of questions in each language for the categories in Yahoo! Answers that are used in the CriES dataset.

diversion, it was important to identify categories with a low share of such questions.

In the challenge, four languages are considered: English, German, French and Spanish. As category names are language-specific, questions and answers from categories corresponding to the selected categories in other languages are also included, for example `Gesundheit` (German), `Santé` (French) and `Salud` (Spanish) that all correspond to the `Health` category.

The selected dataset consists of 780,193 questions, each question having exactly one best answer. The answers were posted by 169,819 different users, *i.e.* potential experts in our task. The answer count per expert follows a power log distribution, *i.e.* 54% of the experts posted one answer, 93% 10 or less, 96% 20 or less. 410 experts published answers in more than one language. These multilingual experts posted 8,976 answers, which shows that they are active users in the portal. The language of questions and answers are distributed over languages as shown in Table V.3.

Topic Selection. The topics we used in the context of the CriES challenge consist of 15 questions in each language — 60 topics in total. As these topics are questions posted by users of the portal, they express a real information need. Considering the multilingual dimension of our ER scenario, we defined the following criteria for the topic selection to ensure that the selected topics are indeed applicable for our task:

International domain: People from other countries should be able to answer the question. In particular, answering the question should not require knowledge that is specific to a geographic region, country or culture. Examples:

```
   Pro: Why doesn't an optical mouse work on a
        glass table?
Contra: Why is it so foggy in San Francisco?
```

Expertise questions: As the goal of our system is to find experts in the domain of the question, all questions should require domain expertise to answer them. This excludes for example questions that ask for opinions or do not expect an answer at all. Examples:

```
   Pro: What is a blog?
Contra: What is the best podcast to subscribe to?
```

We performed the following steps to select the topics:

1. Selection of 100 random questions per language from the dataset (total of 400 candidate topics).

2. Manual assessment of each candidate topic by three human assessors. They were instructed to check the fulfillment of the criteria defined above.

3. For each question the *language coverage* was computed. The language coverage tries to quantify how much potentially relevant experts are contained in the dataset for each topic and for each of the different languages. The language coverage was calculated by translating a topic into the different languages (using Google Translate) and then using a standard IR system to retrieve all the expert profiles that contain at least one of the terms in the translated query. Topics were assigned high language coverage if they matched an average number of experts in all of the languages. In this way we ensure that the topics are well covered in the different languages under consideration but do not match too many experts profiles in each language. This is important for our multilingual task as we intend to find experts in different languages.

4. Candidate questions are sorted first by the manual assessment and then by language coverage. The top 15 questions in each language were selected as topics.

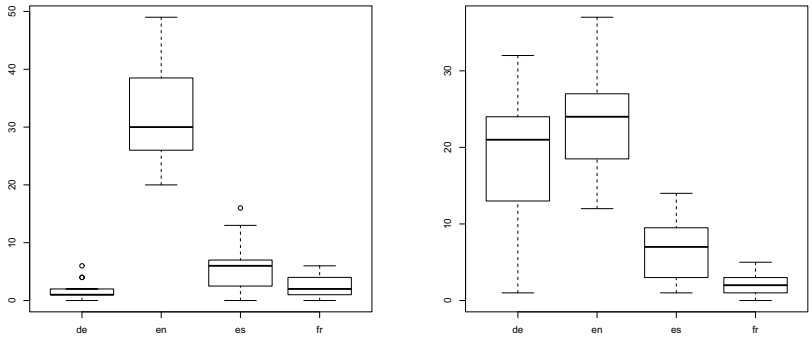
Relevance Assessment. We used result pooling for the evaluation of the retrieval results of the participating groups in the CriES pilot challenge. For each submitted run, the top 10 retrieved experts for each topic were added to the result pool. All experts in this result pool were then judged for relevance in respect to the according topics.

The assessment of experts was based on expert profiles. Assessors received topics and the whole profile of experts, consisting of all answers posted by the expert in question. Based on this information they assigned topic-expert tuples to the following relevance classes:

- 2 — Expert is likely able to answer.
- 1 — Expert may be able to answer.
- 0 — Expert is probably not able to answer.

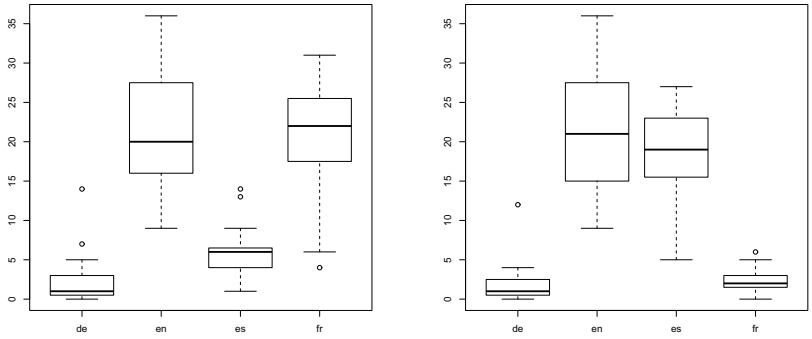
The assessors were instructed to only use evidence in the dataset for their judgments. It is assumed that experts expressed all their knowledge in the answer history and will not have expertise about other topics, unless it can be inferred from existing answers.

Overall, six assessors evaluated 7,515 pairs of topics and expert profiles. The distribution of relevant users for the topics in the four different languages is presented in Figure V.3. In order to visualize the multilingual nature of the task we also classified



(a) English Topics

(b) German Topics



(c) French Topics

(d) Spanish Topics

Figure V.3: Distribution of the numbers of relevant experts for the English, German, French and Spanish topics. Experts are classified to languages based on their answers submitted to the Yahoo! Answers portal. For each set of topics in the four languages and for each expert language, the distribution of the numbers of relevant experts in the according language is visualized using a box plot.

relevant users to languages using their answers in the dataset. The distribution of relevant users for the topics in the four languages is shown separately for each user language group.

The analysis of the relevant user distribution shows that for all topics the main share of relevant users publish answers either in the topic language or in English. This motivates the cross-language expert retrieval task we consider, as monolingual retrieval in the topic language or cross-lingual retrieval from the topic language to English do clearly not suffice. The number of relevant experts posting in a different language than the topic language or English constitute a small share. However, the percentage is too large to neglect these experts — for example Spanish experts for German topics.

V.5.3 Evaluation Measures

The problem of ER is a subtask of IR and therefore allows us to use standard IR evaluation measures. These measures have already been introduced in Section II.5 and a precise definition can also be found in Section IV.5.1 in the context of the evaluation of concept-based retrieval models. We use the following evaluation measures for the ER experiments that can be summarized as follows:

Precision at cutoff rank 10 (P@10): The share of relevant experts in the top 10 retrieved experts.

Recall at cutoff rank 100 (R@100): The share of relevant experts in the top 100 retrieved experts in respect to the total number of relevant experts for a given topic.

Mean Reciprocal Rank (MRR): This measure depends on the rank of the first relevant experts in the ranking. Higher values of MRR imply that these experts have lower ranks on average for all topics.

Mean Average Precision (MAP): In many IR evaluation campaigns, MAP has been used as standard evaluation measure. This is a precision oriented measure that does not depend on a specific cutoff rank. As for most measures, higher values of MAP imply better performance of the retrieval system.

All the measures presented above depend on complete relevance assessments, *i.e.* the relevance of all retrieved experts is known in the evaluation. However this is not always given. The ground truth of retrieval challenges is often built using the relevance assessments of a result pool that contains the retrieval results of several search systems. Using this ground truth to evaluate new runs that were not part of the initial result pool leads to the problem of *incomplete assessments*. Most probably, experts will be retrieved in the new runs that are not in the result pool and therefore it is unknown if they are relevant or not. By assuming that all retrieved experts that have not been judged are not relevant, all measures as presented above can be

applied to evaluate new runs. The problem is that this approach underestimates the performance as these experts might also be relevant. Buckley and Voorhees [2004] suggest to use an alternative evaluation measure that is robust against incomplete relevance assessments:

Binary Preference Measure (BPREF): This measure is based on the order of relevant and non-relevant experts in the ranking. Given topic q and its set of relevant experts R and of non-relevant experts N known from the relevance assessment, the BPREF on a ranking of experts α is defined as:

$$\text{BPREF}_q(\alpha) = \frac{1}{|R|} \sum_{r \in R} 1 - \frac{|\{e \in N \mid \text{rank}_\alpha(e) < \text{rank}_\alpha(r)\}|}{|R|}$$

Intuitively, BPREF measures the number of wrong sequences in all possible pairs of retrieved experts. Thereby, wrong sequence means that a non-relevant expert is retrieved before a relevant expert.

In our experiments, we present results that were not officially submitted to the CriES workshop and were therefore not part of the result pool. By using BPREF as evaluation measure, we ensure a fair comparison of the different systems without having complete relevance assessments for the new runs.

V.5.4 Baselines

We use several baselines in our experiments that are compared to the proposed ER models using combined evidence as presented in Section V.4. In summary, these retrieval baselines can be classified as follows:

Popularity Baseline: In the context of category systems, popularity models as described in Section V.3.2 can be used to determine a priori scores of experts in each category. The assumption is that popular experts in a category are probably able to answer questions related to the topic of the category.

In informed scenarios that assume the knowledge of the target categories of topics, we define the popularity baseline as the ranking of experts according to their popularity in this target category. As we consider different approaches to model the popularity of experts — using the answer frequencies and using the social network of the category — this also results in different popularity baselines.

Probabilistic Retrieval Model: We use a standard probabilistic retrieval model — namely the multilingual extension of BM25 — as baseline approach. We compare to this state-of-the-art retrieval model to illustrate the improvements in retrieval performance achieved by our proposed models.

Standard Language Models: We use the language model for ER proposed by Balog et al. [2009a] as further baseline. This model is described in detail in Section V.2.3. As our task is multilingual ER, we extended this model to support multilingual retrieval as described in Section V.3.1.

CriES Pilot Challenge: Different groups submitted their retrieval results to the CriES pilot challenge. This retrieval challenge is based on the same dataset and topics as used in our experiments, which allows direct comparison to the approaches of these participating groups.

In the following, we further describe the probabilistic retrieval model BM25 and summarize the retrieval models used by the groups that participated at the CriES pilot challenge.

BM25. As experts are defined through textual profiles, the task of retrieving experts can be reduced to a standard document retrieval scenario. As state-of-the-art probabilistic retrieval model to compare to we chose the BM25 model [Robertson and Walker, 1994]. For the multilingual scenario, results from language specific indexes are combined using Z-Score normalization [Savoy, 2005], which has proven to be effective in different multilingual IR challenges, for example the CLEF ad-hoc task.

BM25 + Popularity. Given the informed scenario, the category of a new topic is known. In this case, the popularity of experts in this target category can be combined with any vector space or probabilistic retrieval model by simply adding the popularity to the score of each expert. In our experiments, we will combine BM25 with the frequency based popularity model PM_{freq}. To ensure compatible values, we will again first apply Z-Score normalization to the results of the probabilistic model and the popularity model, resulting in s^*_{BM25} and $\text{PM}^*_{\text{freq}}$. The final score is then defined by the weighted sum of the normalized values:

$$s(e,q) = \alpha s^*_{\text{BM25}}(e,q) + (1-\alpha)\text{PM}^*_{\text{freq}}(e,c_q)$$

In our experiments, we empirically optimized the weight factor α.

CriES Pilot Challenge Models. The retrieval systems used for the submitted runs to the CriES pilot challenge can be classified to the following main retrieval approaches:

MLIR on Expert Profiles: Similar to the probabilistic baseline as defined above, text profiles of experts can be interpreted as documents. This allows to apply MLIR approaches to the task of ER. All groups used query translation based on Machine Translation systems to support multilingual retrieval (see Section II). As retrieval models, different vector space models and probabilistic models were applied.

Matching CriES runs:

- Iftene (run ids: *run0, run1*) [Iftene et al., 2010]
- Leveling (run ids: *DCUa, DCUq*) [Leveling and Jones, 2010]

Exploitation of Social Network: As described in the definition of the popularity models in Section V.3.2, the social network that is based on users, questions, answers and categories can be used to support ER systems. These systems that are based on the social network of experts use the HITS algorithm and the degree of nodes to define scoring functions for experts.

Matching CriES runs:

- Iftene (run ids: *run0, run1, run2*) [Iftene et al., 2010]
- Leveling (run ids: *DCUa, DCUq*) [Leveling and Jones, 2010]

Resource-indexing based Retrieval: Text profiles of experts can be matched to resources using resource-indexing systems based on external knowledge sources. These matches can be used to map experts to interlingual concept spaces similar to Cross-lingual Explicit Semantic Analysis (CL-ESA) as presented in Chapter IV. In contrast to CL-ESA, the approach of Herzig and Taneva [2010] is based on a set of linked resources as concept space. Herzig and Taneva also used manual resource-indexing of topics as their approach depends on larger text snippets for automatic indexing.

Matching CriES runs:

- Herzig (run ids: *1-boe-06-03-01-q01m, 2-boe-06-03-01-q01, 3-boe-07-02-01-q01m*) [Herzig and Taneva, 2010]

V.5.5 Results of Baselines

In Section V.5.4, we defined several baseline retrieval approaches to the problem of multilingual ER. This includes popularity based models as well as runs submitted to the CriES pilot challenge. The results of these baselines are presented in Table V.4.

For the CriES challenge, we chose precision-oriented evaluation measures. Precision at cutoff rank 10 (P@10) represents the average percentage of relevant experts in the top 10 retrieved experts. This measure is motivated by the behavior of people using search engines. Most users only consider the results that are presented on the first result page, which are usually around 10 items. The precision in these top results is therefore most important for the user satisfaction.

In the evaluation, we also use Mean Reciprocal Rank (MRR) as further evaluation measure. This measure is based on the rank of the first relevant expert only. This is motivated by the ER scenario. Assuming that the first relevant experts will be able to satisfy an information need, the rank of this expert is most critical. A MRR

(a) Results measured by precision at cutoff level 10 (P@10) and Mean Reciprocal Rank (MRR). The results are presented for both strict and lenient assessments.

		Strict		*Lenient*	
	Run Id	*P@10*	*MRR*	*P@10*	*MRR*
	Exploitation of the Social Network				
1	Iftene (run2)	.62	.84	.83	.94
2	Popularity model $PM_{pagerank}$	.56	.85	.67	.92
3	Popularity model PM_{freq}	.67	.89	.79	.96
	MLIR on Expert Profiles				
4	Iftene (run0)	.52	.80	.82	.94
5	Iftene (run1)	.47	.77	.77	.93
6	BM25 + Z-Score	.19	.40	.39	.63
	MLIR using the Social Network				
7	Leveling (DCUa)	.09	.14	.40	.51
8	Leveling (DCUq)	.08	.16	.42	.54
	Resource-indexing based Retrieval				
9	Herzig (3-boe-07-02-01-q01m)	.49	.76	.87	.93
10	Herzig (1-boe-06-03-01-q01m)	.48	.77	.86	.94
11	Herzig (2-boe-06-03-01-q01)	.35	.65	.61	.74

(b) The result overlap of the baseline runs. The presented values correspond to the count of retrieved experts of each run for all topics that are also retrieved by the compared run in the top 10 results. The absolute numbers of common experts are presented below the diagonal, relative values in respect to all retrieved experts above the diagonal.

	1	2	3	4	5	6	7	8	9	10	11
1	-	62%	76%	51%	41%	0%	1%	0%	14%	14%	11%
2	374	-	74%	41%	35%	0%	0%	0%	17%	17%	12%
3	457	441	-	45%	37%	0%	0%	0%	17%	17%	12%
4	307	243	271	-	64%	0%	1%	1%	14%	14%	11%
5	248	208	223	385	-	1%	2%	2%	13%	13%	9%
6	1	1	1	1	4	-	10%	2%	1%	1%	1%
7	3	2	2	4	11	58	-	12%	0%	1%	0%
8	1	0	0	3	13	12	72	-	1%	1%	1%
9	82	101	101	83	78	3	2	4	-	92%	46%
10	83	100	102	83	78	3	3	4	554	-	48%
11	67	71	73	63	54	3	1	4	276	286	-

Table V.4: Results of the baseline runs. We present the performance measured by standard evaluation measures and the pairwise overlap of runs in respect to retrieved experts.

value of 1 would be optimal, meaning that all experts on the first rank are relevant for the according query.

All evaluation measures are presented based on strict and lenient assessment. As described in Section V.5.2, the assessment of retrieved experts was based on three classes of relevance. Strict assessment only considers experts in class 2 as relevant — expert that are likely able to answer the given question. Lenient assessment additionally defines experts in class 1 as relevant — expert that may be able to answer the question.

As further statistics of the baseline runs, we present the *result overlap* in Table V.4. This is a pairwise comparison of retrieval runs. The overlap measures the number of experts that are retrieved by both runs in the top 10 results — aggregated for all topics. The overlap matrix in Table V.4 contains the absolute numbers of common experts (below the diagonal) and the relative numbers in respect to all experts in the top 10 results (above the diagonal). In the discussion, we use these result overlaps to visualize the similarities and differences of the different baseline approaches.

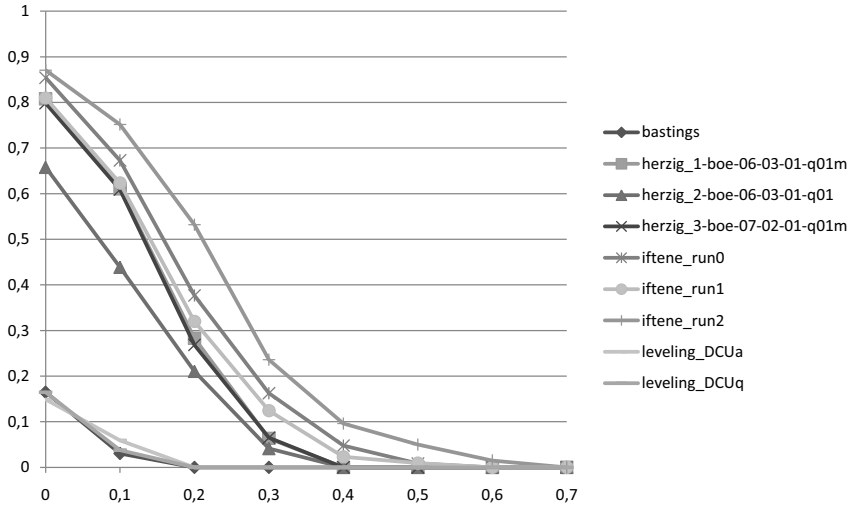
All results in Table V.4 are marked with run ids. These ids will be used in parenthesis in the discussion to refer to the according run. For example (2) refers to the popularity model $\text{PM}_{\text{pagerank}}$.

Evaluation of CriES Runs. Four different groups participated at the CriES pilot challenge. Results of the most successful runs are presented in Table V.4. All submitted runs were used to build the result pool for the manual relevance assessments. In addition to the presented evaluation measures, we also visualize the precision/recall curves of the submitted runs in Figure V.4. In this figure, interpolated recall values for different precision levels are used to draw curves for each run. The values on the X-axis correspond to the precision levels and the values on the Y-axis to the interpolated recall. The *area under the curve* can be used as further evaluation measure that is intuitively visible in Figure V.4.

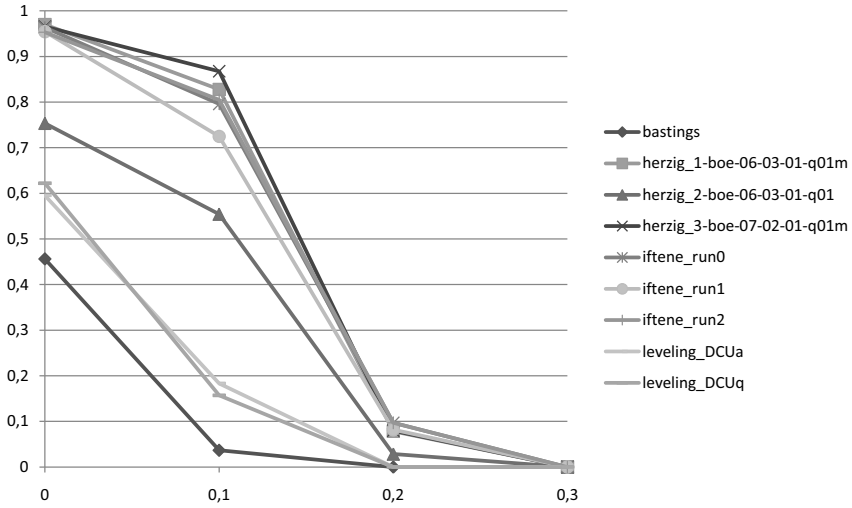
The presented results in Table V.4 and Figure V.4 allow the following observations:

- Most submitted runs beat the probabilistic baseline (6). Exceptions are the runs (7,8), that are much worse than (6) based on strict assessment and comparable to (6) based on lenient assessment. We will not include any further analysis of the retrieval system of runs (7,8) as any relevant approach should be able to improve the simplistic baseline (6).

- The runs solely using the social network have the best performance based on strict assessment (1,3). The best approach measured by P@10 and MRR based on strict assessment is defined by the frequency-based popularity model PM_{freq} (3) in the target category of the query. Run (1) performs slightly worse based on strict assessment but has higher precision than (3) based on lenient assessment. These results show that the social network contains strong evidence for



(a) Results based on strict assessment.



(b) Results based on lenient assessment.

Figure V.4: Precision/recall curves based on interpolated recall.

expertise in the informed scenario.

- The combination of the information given by the social network and by text based retrieval models was not successful (4,5), especially based on strict assessment. For example run (1) has precision of .62 while (4) — which is based on (1) — only achieves precision of .52. We address this problem of combining the different sources of evidence in the mixture language model. Results of this model are presented in Section V.5.6.

- Resource-indexing based retrieval achieves best results based on lenient assessments — run (9) has a precision value of .87. However based on strict assessment, this run only has a precision of .49, which is .18 less than the frequency based popularity model (3).

Overlap of Retrieved Experts. The result overlap of the baseline runs as presented in Table V.4 shows the variety of the different retrieval approaches. We draw the following conclusion based on these overlap values:

- The probabilistic baseline (6) has almost no overlap ($\leq 1\%$) to any of the best performing approaches (1,3,4,9). This means, that the expertise model is completely different as a disjunct set of experts is returned.

- The best performing approaches based on the social network (1,3) are similar with a high overlap of 76%. As the rankings of experts are similar, both approaches seem to exploit the same information given by the number of answers posted by users.

- The combined approach (4) shares half of the retrieved experts with social network based approach (1). This shows that the combined approach introduces a different model of expertise — which is however not able to improve the performance measured by P@10 and MRR based on strict assessment. Based on lenient assessment, similar performance is achieved with this partially overlapping set of retrieved experts.

- The resource-indexing based retrieval approach (9) has low overlap ($\leq 17\%$) with the social network based approaches (1,3) and the combined approach (4). This shows that the model of expertise used in the resource-indexing based approach is different to the model used in the other approaches. Different sources of evidence are exploited by the resource-indexing based approach — which leads to the best performance based on lenient assessment.

V.5.6 Results of the Mixture Language Models

The results of our ER experiments are presented in Tables V.5 and V.6. Table V.5 contains the results on the non-informed scenario. In this case, the category of new

(a) Results based on strict assessment.

Id	*Model*	*P@10*	*R@100*	*MAP*	*BPREF*
	Baselines				
1	BM25 + Z-Score	.19	.10	.04	.08
2	P_{LM} [Balog et al., 2009a]	$^{(1)}$.50	$^{(1)}$.45	$^{(1)}$.22	$^{(1)}$.33
	Mixture Language Model				
3	P_{MLM} $(.1P_{\text{profile}} + .9P_{\text{cat}})$	$^{(2)}$**.55**	$^{(2)}$**.50**	$^{(2)}$**.26**	$^{(2)}$**.37**
	Discriminative Models				
4	DM (Logistic Regression)	.31	.40	.14	.31
5	DM (MLP)	.34	.40	.15	.30
	Re-ranking of run 3 (top 100 experts)				
6	DM (Logistic Regression)	.39	.50	.21	**.37**
7	DM (MLP)	.39	.50	.21	**.37**
8	LRM (J48)	.23	.47	.12	.33

(b) Results based on lenient assessment.

Id	*Model*	*P@10*	*R@100*	*MAP*	*BPREF*
	Baselines				
1	BM25 + Z-Score	.39	.12	.06	.19
2	P_{LM} [Balog et al., 2009a]	$^{(1)}$.69	$^{(1)}$.32	$^{(1)}$.19	$^{(1)}$.30
3	P_{MLM} $(.1P_{\text{profile}} + .9P_{\text{cat}})$	$^{(2)}$**.73**	$^{(2)}$**.36**	$^{(2)}$**.22**	$^{(2)}$**.33**
4	DM (Logistic Regression)	.43	.29	.12	.27
5	DM (MLP)	.48	.29	.13	.27
6	DM (Logistic Regression)	.51	.36	.18	**.33**
7	DM (MLP)	.53	.36	.17	**.33**
8	LRM (J48)	.34	.06	.04	.06

(c) Result overlap measured by common experts of two runs in the top 10 retrieved experts.

	1	2	3	4	5	6	7	8
1	-	2%	2%	1%	3%	1%	2%	0%
2	10	-	77%	22%	23%	34%	35%	5%
3	9	464	-	25%	24%	36%	35%	5%
4	5	131	149	-	42%	58%	38%	5%
5	18	136	146	250	-	39%	59%	4%
6	8	202	214	347	233	-	58%	5%
7	13	207	210	230	354	346	-	4%
8	1	27	29	27	21	28	23	-

Table V.5: Non-informed experiments: Expert retrieval results on the Yahoo! Answers dataset with unknown category of new questions. Precision at cutoff level 10 (P@10), recall at cutoff level 100 (R@100), Mean Average Precision (MAP) and BPREF are used as evaluation measures.

(a) Results based on strict assessment.

Id	*Model*	*P@10*	*R@100*	*MAP*	*BPREF*
	Baselines				
9	PM_{freq} (Popularity Model)	$^{(3)}$.67	.43	.29	.37
10	BM25 + PM_{freq} + Z-Score	$^{(3,6,8)}$**.71**	$^{(6)}$.47	$^{(3,6)}$.31	$^{(3,6)}$.39
	Mixture Language Model				
11	P_{MLM} ($.5P_{profile} + .5PM_{freq}$)	$^{(3)}$.67	$^{(6)}$.47	$^{(3,6)}$.31	$^{(3,6)}$.39
	Discriminative Models				
12	DM (Logistic Regression)	.50	.54	.27	.39
13	DM (MLP)	.60	$^{(3,8)}$**.60**	$^{(3,8)}$**.33**	$^{(3,8)}$**.43**
	Re-ranking of run 8 (top 100 experts)				
14	DM (Logistic Regression)	.53	.47	.25	.37
15	DM (MLP)	.62	.47	.31	.39
16	LRM (Logistic Regression)	.55	.46	.26	.37
17	LRM (J48)	.58	.47	.27	.39

(b) Results based on lenient assessment.

Id	*Model*	*P@10*	*R@100*	*MAP*	*BPREF*
9	PM_{freq} (Popularity Model)	.79	.33	.21	.31
10	BM25 + PM_{freq} + Z-Score	$^{(3,6)}$**.84**	$^{(6)}$.37	$^{(3,6)}$.25	$^{(3,6)}$.35
11	P_{MLM} ($.5P_{profile} + .5PM_{freq}$)	.80	$^{(6)}$.37	$^{(3,6)}$.24	$^{(3,6)}$.35
12	DM (Logistic Regression)	.69	.41	.24	.37
13	DM (MLP)	.78	$^{(3,8)}$**.45**	$^{(3,8)}$**.30**	$^{(3,8)}$**.42**
14	DM (Logistic Regression)	.70	.37	.22	.35
15	DM (MLP)	.79	.37	.26	.35
16	LRM (Logistic Regression)	.74	.36	.23	.34
17	LRM (J48)	.73	.37	.23	.35

(c) Result overlap measured by common experts of two runs in the top 10 retrieved experts.

	9	10	11	12	13	14	15	16	17
9	-	92%	90%	41%	58%	51%	68%	52%	60%
10	550	-	90%	41%	61%	52%	71%	53%	62%
11	539	541	-	42%	61%	52%	71%	53%	61%
12	245	247	253	-	49%	68%	45%	60%	39%
13	348	363	364	294	-	49%	78%	49%	53%
14	305	309	313	407	292	-	58%	82%	49%
15	408	424	426	268	466	347	-	59%	63%
16	310	318	319	360	291	489	354	-	50%
17	359	370	368	231	315	292	379	299	-

Table V.6: Informed experiments: Expert retrieval results on the Yahoo! Answers dataset using informed approaches that use the category of new questions. Precision at cutoff level 10 (P@10), recall at cutoff level 100 (R@100), Mean Average Precision (MAP) and BPREF are used as evaluation measures.

questions is not known. Table V.6 contains the results on the informed scenario using a priori knowledge of the category of new questions.

As evaluation measures, we use precision at cutoff rank 10 (P@10), recall at cutoff rank 100 (R@100), Mean Average Precision (MAP) and binary preference ranking (BPREF). Not all runs presented in Tables V.5 and V.6 were part of the result pool in the context of the CriES challenge. The values for P@10, R@100 and MAP are therefore underestimated. To address this problem, we include BPREF as evaluation measure that has been designed to be robust against incomplete assessments.

Similar to the results presented in Section V.5.5, we also present the pairwise result overlap of all runs in Tables V.5 and V.6. This will again be used to identify approaches that are based on similar models of expertise.

The statistical significance between systems was verified using a paired t-test at a confidence level of .01. Significant results are marked with the id number of the system compared to. For example, the P@10 value .50 of the language model P_{LM} (2) shows a statistical significant difference compared to the P@10 value .19 of the probabilistic model BM25 (1) in Table V.5.

In the following, we discuss the results of the baseline runs, the mixture language model (MLM), the discriminative model (DM) and the Learning to Rank model (LRM). We will use the system id in parenthesis to link to the according results in Tables V.5 and V.6.

Informed vs. Non-informed Scenario. As a first observation, it is important to mention that the usage of the information given by the topic category generally improves retrieval results (results in Table V.5 vs. results in Table V.6). As example, the probabilistic baseline (1) has a precision of .19 while the linear combination of this baseline with the frequency based popularity model results in a substantially improvement to a precision of .71. Similar effects can be seen for the discriminative model. Using the same Machine Learning model and the same retrieval approach, precision can be improved from .34 to .60 by adding features that exploit the target category of queries.

However, analyzing approaches that do not exploit the target category of queries is still important as there are use cases that correspond to the non-informed scenario. An example are questions that could be assigned to several categories and the selection of a specific category is therefore difficult for users of a SQAS.

Mixture Language Model. The mixture language model that combines language models of user profiles and categories shows high performance in the non-informed scenario. We make the following observations based on the results presented in Table V.5:

- The probabilistic multilingual retrieval system (1) has poor performance compared to the mixture language model (3). For example based on strict assess-

ment, P@10 is improved from .19 to .55. Similar improvements are given for the other evaluation measures.

- The reference language model (2) introduced by Balog et al. [2009a] is outperformed by the mixture language model. The statistical significance of this improvement is given for all used evaluation measures. For example based on strict assessment, MAP is improved by .04 and R@100 by .05.

These results show that the mixture language model is able to successfully combine the different sources of evidence in the non-informed scenario. By considering the category language models in the retrieval model, the results are improved in respect to the reference approaches.

In the informed scenario, the mixture language model (11) outperforms the baseline using the frequency based popularity model (9). Improvements with statistical significance are given for R@100, MAP and BPREF. For example based on lenient assessment, MAP is improved by .03 and BPREF by .04. However, the linear combination of the probabilistic model and the popularity model (10) achieves better precision having the same values for R@100, MAP and BPREF compared to the mixture language model (11). For example based on strict assessment, P@10 is improved from .67 to .71. This shows that the simple linear combination of the different sources of evidence is superior to the mixture language model in the informed scenario.

Discriminative Models. The discriminative models in the non-informed scenario (4,5) are not able to compete against the approaches based on language models. In fact, both approaches are consistently worse than the reference language model (2) for all evaluation measures. We are not able to give a satisfying explanation of this observation. Our hypothesis is that the features in the non-informed scenario do not allow to distinguish between relevant and non-relevant experts.

Considering the informed scenario, the discriminative model (13) outperforms all other models in respect to R@100, MAP and BPREF. We make the following observations based on the results presented in Table V.6:

- Based on both strict and lenient assessments, the discriminative model using MLPs (13) achieves best results in respect to R@100, MAP, BPREF. For example based on strict assessment, BPREF of .37 of the popularity model is improved to .43. The improvement of R@100 is even higher, the results of the mixture language model (11) are improved by .13.

- Using the discriminative model, the precision measured by P@10 is worse compared to the baselines — the popularity model (9) and the linear combination of the probabilistic and the popularity model (10). The discriminative model is therefore optimizing recall, which is substantially improved. However, other measure of precision — MAP and BPREF — are also improved, which shows that the discriminative approach also has adequate precision.

- MLPs are the best choice for the underlying ML system used in the discriminative model. Using Logistic Regression deteriorates the results, for example by .06 for R@100 based on strict assessment.

Re-ranking of Results. The re-ranking of the top 100 experts that are found using the mixture language models (3,11) does not show any obvious improvement in both non-informed and informed scenarios. This holds for the discriminative approaches (6,7,14,15) and for the Learning to Rank approaches (8,16,17). In fact, the results are worse compared to the original rankings (3,11) that are used for the re-ranking.

An interesting fact is that the drop of performance is not observed for BPREF. In the non-informed scenario, the re-ranking approaches using discriminative models (6,7) have same BPREF than the original model (3). In the informed scenario, this also holds for the Learning to Rank approach (17) using the J48 decision tree compared to the original informed model (11). All runs of re-ranking approaches were not part of the result pool that has been assessed in the context of CriES. Therefore, the results of the re-ranking approaches are probably influenced by incomplete assessments. This is supported by the observation that the BPREF values do not show the same deterioration. For realistic judgments of the re-ranking approaches, complete relevance assessments of all retrieved experts are needed which remains future work of this thesis.

Diversity of Retrieved Experts. The results presented in Table V.6 show that using the popularity model is a strong baseline in the informed scenario. In fact, this is one of the most successful retrieval strategies measured by the evaluation measures. In the ER scenario in SQASs, this means that users with many posts in a category are the obvious experts that are able to answer new questions in this category.

However in a realistic scenario, these users will not able to answer all new questions in a specific category. An alternative evaluation measure could be defined by the *diversity of retrieved experts*. ER should not only identify relevant experts but also have a diverse result set that potentially distributes new questions to many different users.

We propose to measure the diversity using the result overlap to the popularity model (9) as presented in Table V.6. High overlap means that only the popular experts are retrieved and that the diversity is low. The results show that the probabilistic model (10) has a high overlap of 92% — the set of retrieved experts is almost identical to the popularity baseline (9). The same holds for the mixture language model with an overlap of 90%. The discriminative model (13) has much lower overlap of 58%. This means that the diversity of retrieved experts is higher at the same time as having improvements of R@100, MAP and BPREF. This is a further argument to favor the discriminative approach in the ER scenario.

Correctly Classified Instances	*Features*
78.8%	AVG($[P_{\text{profile}} \times \text{PM}_{\text{freq}}]$)
79.5%	+MIN($[\text{PM}_{\text{freq}}]$)
79.8%	+MEDIAN($[P_{\text{cat}}]$)
80.1%	+AVG($[P_{\text{profile}} \times P_{\text{profile+c}} \times \text{PM}_{\text{freq}}]$)
80.2%	+DLH13 + Z-Score
80.3%	+MEDIAN($[P_{\text{cat}} \times P_{\text{term}}]$)
80.3%	+DEVIATION($[P_{\text{cat}} \times \text{PM}_{\text{freq}}]$)
80.4%	+MAX($[P_{\text{profile}} \times P_{\text{profile+c}} \times P_{\text{cat}}]$)
80.4%	+MEDIAN($[P_{\text{profile}}]$)
80.4%	+MEDIAN($[P_{\text{profile}} \times \text{PM}_{\text{freq}}]$)

Table V.7: Top-10 ranked list of features used for the discriminative model in the informed scenario. Features were selected by a greedy stepwise algorithm. Starting with an empty set of features, the feature resulting in the most correctly classified instances was added in each step. Thereby, the classification was based on a logistic regression classifier using the current set of selected features as input.

V.5.7 Feature Analysis

The performance of the discriminative model is based on the performance of the underlying ML model. This model uses the features defined in Section V.4 to build a classifier based on the training data. An interesting question is the importance of each feature in the ML model.

To get an idea of which features contribute most to the success of the classifier function, we performed a feature analysis of the features used in the discriminative model. Features were evaluated by their performance in the classification. The feature set was selected by a greedy stepwise algorithm. The results are presented in Table V.7, sorted by the share of correctly classified instances that was achieved by adding each feature. We derive the following results from this analysis:

- The top features are based on the category of the question (PM_{freq} and $P_{\text{profile+c}}$). This was expected due to the high performance of the popularity model.

- All sources of evidence are present in the top features: User profiles (P_{profile} and $P_{\text{profile+c}}$), categories (P_{cat} and PM_{freq}), background language model (P_{bg}) and standard MLIR models (DLH13 + Z-Score). This shows that each source adds information that is useful for the classification.

- Considering dependencies on term level of the different sources of evidence is important, as many top ranked features are indeed products of single-evidence term probabilities which capture this dependency, for example $[P_{\text{profile}} \times \text{PM}_{\text{freq}}]$.

V.6 Summary of Results

In the following, we give a summary of all results that were presented in this chapter. This includes the results of our experiments, our main contributions and also our lessons learned.

V.6.1 Results of the Experiments

The main results of our experiments on the Expert Retrieval scenario using the Yahoo! Answers dataset can be summarized as follows:

Informed vs. Non-informed Scenario. The results of the different retrieval models are not consistent across the two scenarios that we consider — the non-informed scenario and the informed scenario that requires former knowledge about the target category of topics. Some of the top performing models in one scenario do not achieve good results in the other scenario. For example, the discriminative model is not improving results in the non-informed scenario compared to the mixture language model — but is one of the best models in the informed scenario. Our conclusion is that there is no single solution for both scenarios. Different retrieval models are needed that are optimized to the according retrieval task.

Mixture Language Model exploits Category System. The mixture language model is one of the best performing models in both scenarios. In particular, it outperforms the baseline language model that is not using the categorical information. Our conclusion is that the mixture language model, which we defined on expert and category profiles, is able to exploit the knowledge that is given by the category system in Yahoo! Answers to successfully improve the retrieval results.

Popularity Model. The popularity model based on answer frequencies is a strong baseline that outperforms all submitted runs to the CriES challenge when using strict assessment. For lenient assessment, only two retrieval systems are able to improve this model in respect to some evaluation measures. These systems are based on an alternative approach to exploit the social network as well as on resource-indexing. In our experiments, the mixture language model and the discriminative approach outperform the popularity baseline which is indicated by all used evaluation measures.

Discriminative Model in Informed Scenario. The discriminative model that we propose in this chapter is able to further improve the results of the mixture language model in the informed scenario. As the features that are used in the discriminative model are based on the same language models, this shows that the combination of different sources of evidence can be optimized using Machine Learning techniques. Further, the results obtained using the discriminative model have more diversity of retrieved experts compared to the other models that are mostly dominated by the

popularity baseline. In the question routing use case, this means that the payload of answering questions is distributed to a larger set of experts.

V.6.2 Lessons Learned

In the following, we discuss further considerations about the presented experiments and analyze some negative results. We think that these results might be helpful to avoid mistakes and dead ends in future research.

Discrete Classifier in Discriminative Approach. Using classifiers with discrete output such as decision trees resulted in poor retrieval results in the discriminative approach. In this case, many experts get the same or very similar scores — which makes them indistinguishable for ranking purposes. Cross-validation of the underlying ML model on the training set was usually comparable to regression functions such MLPs or Logistic Regression, but differences were huge when applying the classifier to the actual retrieval task. For example, the re-ranking approach in the informed scenario based on the J48 decision tree results in a substantial performance drop compared to the mixture language model.

Training Data for ML Models. The discriminative model was only effective when training on strict relevance judgments, irrespective of the fact whether evaluation was performed in strict or lenient mode. We also performed experiments training on pairs of questions and expert providing the best answer as specified in the Yahoo! Answers dataset (*i.e.* not using the relevance judgments provided in the CriES dataset). Let's call these pairs *best-answer-pairs*. This did also yield unsatisfactory results. A possible explanation for the negative performance is the fact that negative examples where randomly sampled from *non-best-answer-pairs*, thus leading to a noisy dataset as many experts might be relevant for a given question in spite of not having provided the best answer.

Bias of Relevance Assessments. Our experiments show that the popularity model is a very strong baseline that achieves high evaluation values when applied as retrieval model. This means that experts with many associated documents are more likely to give answers to new questions. While this might be true in the considered SQAS, there might also be the affect of a bias in our relevance assessments. The human assessors judged the text profile of experts which are possibly long documents for users who posted many answers. These profiles contain questions and answers of many different topic fields and are hard to overlook. The assessors might have been drawn to judge these users as relevant. This could be the explanation of the possible bias towards active users.

To evaluate this hypothesis, alternative relevance assessments are required. By assessing single question/answer pairs, the bias towards long profiles could be

avoided and this could be compared to the existing relevance assessments. However, this introduces new problems. In some cases, the expertise might be defined in the context of several answers, which is not covered in this type of evaluation.

Chapter VI

Enriching the Cross-lingual Structure of Wikipedia

Wikipedia is a rich knowledge source that contains descriptions of concepts in a broad range of topic fields. In Chapter IV, we presented CL-ESA as an approach that allows using the concepts provided by Wikipedia to define a concept-based document representation. Other applications of the knowledge encoded in Wikipedia are presented in Chapter III. In addition to the concept descriptions, Wikipedia also aligns articles and categories across languages. We exploited these alignments in Chapter IV to define an interlingual concept space that supports Cross-lingual IR and Multilingual IR.

The cross-lingual structure of Wikipedia is based on *cross-language links*. These links are created by adding references to corresponding articles or categories in other languages to the source text of articles or categories. This is manually done in a collaborative process by the Wikipedia editors. Therefore, the existence and correctness of such cross-language links is not guaranteed. Further, it is difficult for editors to create such links, as this requires language skills in at least two different languages. While many editors might be able to create links to international languages such as English, this is certainly not the case for languages that are not widely distributed such as Albanian.

Cross-language links are the only connecting elements of Wikipedia content across languages. Therefore, they are crucial for any application that depends on the alignment of articles or categories in different languages. This also means that the performance of these applications is affected by missing or incorrect cross-language links. In this chapter, we propose an automatic approach to identify missing cross-language links. Our motivation is based on the assumption that enriching the cross-language structure of Wikipedia will also benefit all systems that exploit this structure.

Our approach aims at supporting human editors by suggesting *cross-language*

link candidates. We automatically generate recommendations of link candidates. However, these links are not added immediately to the according Wikipedia articles. Automatic editing of content in Wikipedia is problematic as the introduction of errors will not be accepted by the Wikipedia community. Instead, we suggest new links to the editors which allows them to manually check the correctness while still reducing the effort of inserting new cross-language links. The editors do not have to browse all articles to find possible missing links but are guided through the identified link candidates. This will also decrease the probability that articles or categories with missing links are overseen by the editors. Additionally, the editors do not have to manually identify the target articles or categories of new links as these are also suggested by our automatic approach.

In the following, we will first present the motivation for our approach which detects missing cross-language links in Wikipedia based on statistics of existing links. Then, we describe our classification model that is based on specific features to identify new links. This approach is evaluated using a set of randomly selected Wikipedia articles. Finally, we discuss the self-correctiveness of Wikipedia using the example of our generated list of missing cross-language links. The results of this discussion allow different conclusions about the quality and future of Web 2.0 resources such as Wikipedia.

VI.1 Motivation

In Chapter III, we presented different approaches that exploit the cross-language links in Wikipedia for Semantic Relatedness measures or for cross-lingual or multilingual IR. Cross-lingual Explicit Semantic Analysis with Wikipedia as background knowledge, as presented in Chapter IV, is another example for a model that is based on cross-language links. The performance of these approaches depends on the cross-language link structure of Wikipedia, which should be consistent and needs to have enough coverage.

In the context of this chapter, we have chosen the German and English versions of Wikipedia and computed statistics about the German/English cross-lingual structure to get a clear picture about its consistency and coverage. These findings motivate our approach to learn new cross-language links in Wikipedia.

VI.1.1 Statistics about German/English Cross-Language Links

Our analysis of the cross-lingual structure of Wikipedia is based on English and German Wikipedia dumps from October 2007. We only used articles in the default namespace excluding redirect pages.[1] As preprocessing step, we resolved all links ending in redirect pages to the corresponding articles.

[1] Redirect pages are used to disambiguate different surface forms, denominations and morphological variants of a given unambiguous named entity or concept to a unique form or identifier. In Wikipedia, these pages have no content and directly forward to the corresponding article.

For our analysis, we counted the number of articles in the English and German Wikipedia databases and the total number of cross-language links in those articles that are pointed either to the English or to the German Wikipedia.[2] These links can be classified in the following classes:

Bidirectional: A bidirectional link ll_α that connects article a in Wikipedia database W_α to article b in Wikipedia database W_β has an equivalent counterpart ll_β located at the target article, *i.e.* ll_β connects article b to article a.

No Backlink: A cross-language link with no backlink does not have a counterpart. Given such a link ll that connects article $a \in W_\alpha$ to article $b \in W_\beta$, there is no link located at b pointing to any article in W_α.

Inconsistent Backlink: In this case, the target article of a cross-language link is linked to another article as the source article. Formally, given such a link ll_α that connects article $a \in W_\alpha$ to article $b \in W_\beta$, there is another cross-language link ll_β located at b that connects b to article $a' \in W_\alpha$ with $a' \neq a$.

An example is the English article `3D rendering` that is linked to the German article `3D-Computergrafik`. This German article is linked to the English article `3D computer graphics`, which is a different article than `3D rendering`.

The three classes of cross-language link are analogously defined for links between categories instead of articles.

All the results of our analysis of the cross-lingual structure between the English and German Wikipedia snapshots are presented in Table VI.1. The results show that only a small fraction (14%) of articles in the English Wikipedia was linked to articles in the German Wikipedia. The fraction of German articles linked to English articles was much bigger, but at 45.9% it was still less than half of all articles in the German Wikipedia. For some articles, there may not be a corresponding article in another language due to the country specific content. However, half of all articles in the German Wikipedia have no cross-language link to the English Wikipedia. As most probably not all of them are specific to Germany and have an existing counterpart in the English Wikipedia, there is still a big margin to learn new meaningful cross-language links.

As the fraction of bidirectional links was around 95% in the English and German Wikipedia, high consistency of cross-language links can be taken for granted. This motivates their use in a bootstrapping manner to find new cross-language links.

[2]Cross-language links in Wikipedia are located in the text of the source article or category by a reference to the title of the target article or category. As titles are used as identifiers in Wikipedia, these references correspond to directed links that are well-defined as they connect exactly two articles or categories.

(a) Number of articles.

	Articles
English Wikipedia	2,293,194
German Wikipedia	703,769

(b) Number of cross-language links.

	Links	*Share of Articles*
English → German (EN-DE)	321,498	14.0%
German → English (DE-EN)	322,900	45.9%

(c) Classification of existing cross-language links.

	EN-DE LLs		*DE-EN LLs*	
Bidirectional	303,684	94.5%	303,684	94.1%
No backlink	9,753	3.0%	12,303	3.8%
Inconsistent Backlink	7,845	2.4%	6,132	1.9%

Table VI.1: Statistics about the cross-lingual structure of the English and German Wikipedia (snapshot of September 2009). Only cross-language links (LLs) from English to German articles and vice versa are considered.

VI.1.2 Chain Link Hypothesis

One problem in learning new cross-language links between the German and English Wikipedia is the large number of articles (see Table VI.1). It will surely not be possible to use a classifier on all article pairs, therefore a pre-selection of candidate articles seems appropriate.

In addition to existing cross-language links, our approach to identify new cross-language links also uses *pagelinks* in Wikipedia:

Definition VI.1 (Pagelink) *In a Wikipedia database W,* pagelinks *(PLs) are links between articles in W. They are located at the source article by a reference to the title of the target article.*

In order to preselect a number of relevant articles, we rely on the *chain link hypothesis*. This hypothesis builds on the notion of a chain link:

Definition VI.2 (Chain Link) *For two Wikipedia databases W_α, W_β in corresponding languages α, β, a* chain link (CL) *between two articles $a_\alpha \in W_\alpha$ and $a_\beta \in W_\beta$ is defined by the following link structure:*

$$a_\alpha \xrightarrow{pl} b_\alpha \xrightarrow{ll} b_\beta \xleftarrow{pl} a_\beta$$

or

$$a_\alpha \xrightarrow{pl} b_\alpha \xleftarrow{ll} b_\beta \xleftarrow{pl} a_\beta$$

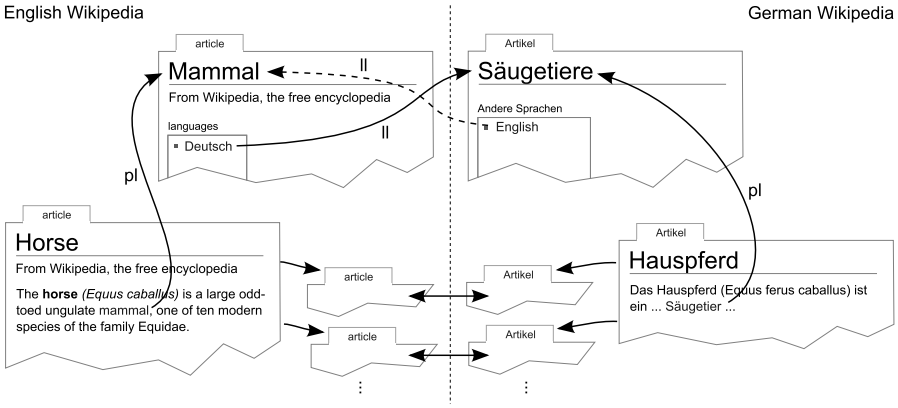


Figure VI.1: Visualization of a chain link that is used to find candidate pages for new cross-language links. The presented articles are linked via pagelinks (pl) and cross-language links (ll).

with $b_\alpha \in W_\alpha$ and $b_\beta \in W_\beta$. Pagelinks between articles are displayed as $\xrightarrow{pl}$ and cross-language links between articles in different languages as $\xrightarrow{ll}$. The articles b_α and b_β are called chain link intermediate articles (CLIAs).

An example for such a chain link between an article in the German Wikipedia and an article in the English Wikipedia is illustrated in Figure VI.1. The article $a_\alpha =$ `Horse` in the English Wikipedia is connected through the displayed chain link to the article $a_\beta =$ `Hauspferd` in the German Wikipedia. The articles $b_\alpha =$ `Mammal` and $b_\beta =$ `Säugetiere` are CLIAs of this chain link that is formed by the pagelink from `Horse` to `Mammal`, the cross-language link from `Mammal` to `Säugetiere` and the pagelink from `Hauspferd` to `Säugetiere`.

Based on chain links, we formulate the chain link hypothesis, which is the basic hypothesis that lays the foundation for the selection of cross-language link candidates:

Definition VI.3 (Chain Link Hypothesis) *Every article is linked to its corresponding article in another language through at least one chain link.*

In order to empirically verify the plausibility of the above hypothesis, we have generated the RAND1000 dataset containing 1000 random articles of the German Wikipedia with existing cross-language links to the English Wikipedia. For all articles in the RAND1000 dataset, we tested if the hypothesis is indeed fulfilled. In particular, for an article a_α in the dataset, connected to the article a_β in the English Wikipedia through a cross-language link, we checked if a_β is in the *candidate set* $\mathcal{C}(a_\alpha)$. The candidate set of an article a_α is composed of all articles that are connected to a_α through at least one chain link.

Candidates	*Percentage*
Full candidate set	95.7%
Restricted candidate set	86.5%

Table VI.2: Percentage of articles in the RAND1000 dataset for which the chain link hypothesis is fulfilled when considering the full candidate set and the restricted candidate set.

However, we noticed that the average number of articles in each candidate set is still not small enough. In case of the RAND1000 dataset, the mean size of the candidate set is 153,402. This means that an approach to find a cross-language link for an article a, which considers all articles in $\mathcal{C}(a)$ as potential candidates, can be very expensive from a computational point of view.

Thus, we also consider a reduction of the number of candidates. Therefore, we define the *support* of a candidate c with respect to an article a in the dataset as the number of existing chain links between a and c. For each article a, we limit the number of candidates to less than 1000 by raising the minimal support of candidates for this article until less than 1000 candidates were selected. This minimal support will be referred to as *support threshold* in this chapter. For each article, we call the set of candidates with higher support than the threshold the *restricted candidate set* $\mathcal{C}'(a)$, which is confined to at most 1000 candidates.

Table VI.2 contains the percentage of articles for which the chain link hypothesis is fulfilled. The results show that for more than 95% of the articles in the RAND1000 dataset the corresponding article in the English Wikipedia is included in the full candidate set. Regarding the restricted candidate set, the hypothesis holds for more than 86% of the articles. With respect to the decrease in performance time by processing at most 1000 instead of 153,402 candidate articles for each source article on average, it seems a good trade-off in terms of best case accuracy.

The chain link hypothesis is therefore strongly supported by this evaluation on the RAND1000 dataset, even after restricting the candidate set to at most 1000 candidates for each article. Based on these findings, the usage of the chain link hypothesis to restrict the set of candidate articles for new cross-language links seems to be promising. The approach presented in the remainder of this chapter strongly relies on the chain link hypothesis as a feature for training a classifier, which is able to predict whether a pair of articles in two languages (German/English in our case) should be connected via a cross-language link. Having motivated our approach and the underlying hypothesis empirically, we describe the approach in more detail in the next section.

VI.2 Classification-based Approach

The main idea behind our approach to learn new cross-language links is to train a classifier, which is able to predict whether a pair of articles (a, b) of two distinguished Wikipedia databases W_α, W_β with $a \in W_\alpha$ and $b \in W_\beta$ should be linked. As it is not feasible to apply the classifier to all article pairs in two languages, we only consider the restricted candidate set $\mathcal{C}'(a) \subset W_\beta$ for an article a as potential targets of a new cross-language link.

We used the popular Support Vector Machine (SVM) implementation *SVMlight* by Joachims et al. [1999] with a linear kernel function as classifier. The classifier is trained with a number of features, which we describe in details below. Features are defined on article-candidate pairs $(a, c) \in W_\alpha \times W_\beta$ with $c \in \mathcal{C}'(a)$ and are based on different sources of evidence. Based on our chain link hypothesis, the support of c with respect to a — defined above as the number of chain links between these articles — is considered and the link structure of the CLIAs is exploited. In addition, the categories of a and c are also considered. As categories are also linked by language links, it is possible to align categories across languages. Finally, we also use simple features based on the title and text of articles.

VI.2.1 Feature Design

The features can be classified in two classes: graph-based and text-based features. Graph-based features exploit the link structure in Wikipedia, which is given by pagelinks, category links and cross-language links. Text-based features are based on the title and text of the Wikipedia articles.

For the definition of graph-based features, we need to specify the number of *inlinks* of an article. Inlinks of an article $a \in W_\alpha$ are pagelinks from another article that are targeted to a. The number of inlinks of a is therefore defined as $\text{INLINKS}(a) = |\{b \in W_\alpha \mid b \xrightarrow{pl} a\}|$.

For the definition of text-based features, we introduce the *Levenshtein Distance* [Levenshtein, 1966], which is a string metric based on the edit distance between two strings. The edit distance is defined as the minimal number of insert, delete and replace operations that is needed to transform one string to another. We use a version of the Levenshtein Distance that is normalized by the string lengths.

As described above, features are based on article-candidate pairs. In the following, we denote the source article as a and the candidate target article as c with $c \in \mathcal{C}'(a)$.

Graph-based Features:

Feature 1 (*Chain Link Count Feature*)
This feature is equal to the support of c with respect to a.

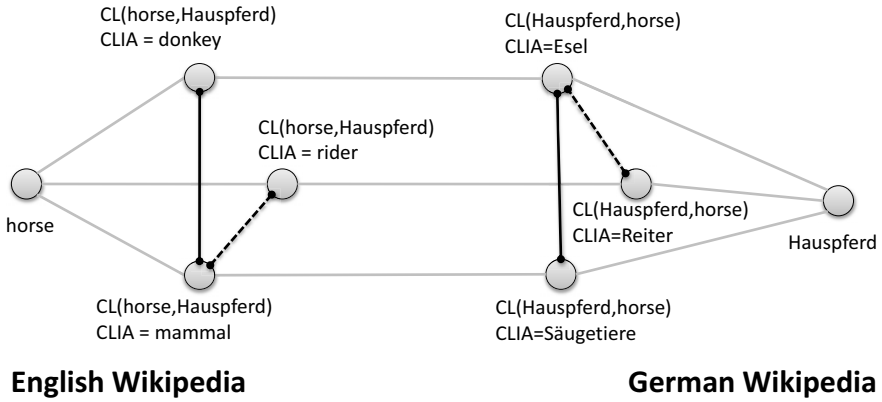


Figure VI.2: Example of two CLIA graphs on a set of chain links in the English and German Wikipedia. Both graphs have a common edge connecting the CLIAs `mammal` and `donkey`, but also non-common edges (`mammal`,`rider`) and (`rider`,`donkey`).

Feature 2 (*Normalized Chain Link Count Feature*)
This feature is the value of Feature 1 normalized by the support threshold that was used to restrict the candidate set for a.

Featureset 3 (*Chain Link Inlink Intervals*)
Given an article a and a candidate c, we compute all the chain links between them. Then, we classify the CLIAs of these chain links to 20 intervals defined over the number of inlinks of each CLIA, i.e. we classify a CLIA b into a bucket according to the value INLINKS(b). Thus, we yield 20 features corresponding to the 20 intervals.

The motivation behind this classification of CLIAs is the assumption that chain links containing CLIAs with fewer inlinks are more specific for a topic and therefore more important for the choice of the correct article. The design of several features corresponding to the different classes of chain links according to the number of CLIA inlinks allows the classifier to explore this assumption. For example, the number of chain links with few CLIA inlinks is a single feature, which gets appropriate weight in the classification model according to the training data.

Feature 4 (*Common Categories Feature*)
The number of categories of candidate article c that are directly linked via a cross-language link to a category of article a.

Feature 5 (*CLIA Graph Feature*)
This feature is based on a similarity measure on graphs. Given two graphs G_α and G_β on the same set of vertices, the similarity is defined as

the number of common edges of these graphs normalized by the number of vertices. For article a and candidate c, the graphs G_α and G_β are defined by the set of chain links between a and c. Thereby, chain links are the vertices in these graphs. Edges in G_α between two chain links exist if the CLIAs in W_α of these chain links are linked by a pagelink in W_α. Analogously, edges in G_β between two chain links exist if the CLIAs in W_β of these chain links are linked by a pagelink in W_β. The CLIA graph feature is given by the value of the defined similarity measure between G_α and G_β.

An example for two CLIA graphs based on a set of chain links is given in Figure VI.2.

Text-based Features:

Feature 6 (*Editing Distance Feature*)

This is the normalized Levenshtein Distance on the titles of the candidate articles pair.

Intuitively, articles with identical titles in the different Wikipedia databases should be identified using this feature. The Levenshtein Distance allows for non-exact matching, which will give high similarity scores to close matches such as "Munich" to "München".

Feature 7 (*Text Overlap Feature*)

This feature computes the text overlap between the text of the candidate article pair. To remain independent of lexical resources, there is no translation involved. In our case, the text of English articles is directly matched to the text of German articles. This feature is useful if the articles for example share many named entities that have identical names or labels in several languages.

VI.3 Evaluation

The evaluation is based on the RAND1000 dataset. As described above, this dataset consists of 1000 articles of the German Wikipedia, each with an existing cross-language link to an article in the English Wikipedia. This set of articles is fixed for all experiments, thus the random selection was only performed once.

In the following, we first analyze this dataset to get a lower bound for the classification experiment. Afterwards, we describe the experimental setup. Finally, we present further results on articles without an existing cross-language link.

VI.3.1 Baseline

In order to find a lower bound for recall, we define a simple method to find language links by matching the titles of articles. The recall of this method on the RAND1000

dataset is equal to the percentage of German articles that are linked to English articles with identical title. The analysis of the RAND1000 dataset showed that 47.0% of the articles in this dataset are linked to English articles with identical title. The reason for this high share is the fact that many Wikipedia articles describe named entities and thus have the same title in different languages. This value defines a lower bound for recall as this method to find new language links is very simple and straightforward. Any other method should exceed the results of this baseline.

VI.3.2 Evaluation of the RAND1000 Dataset

In the experiments, we used a random 3-1-split of the RAND1000 dataset. The first part, containing 750 articles, was used for training the classifier. The remaining 250 articles were used for the evaluation.

In order to evaluate the correctness of our approach, we consider the TOP-k with $k \in \{1, \ldots, 5\}$ candidates. The ranking of candidates is based on the directed distance of their feature vector to the SVM-induced hyperplane. The larger the distance, the higher is the classifier's certainty that the candidate belongs to the positive class, *i.e.* the candidate should be linked to the source article. Hereby, we do not distinguish candidates having feature vectors with a positive distance to the hyperplane and others with a negative distance. Thus, it is possible that some of the candidate articles that appear at the top of the ranking are actually classified to the negative class.

TOP-k Evaluation. As quality measure for the TOP-k evaluation we defined TOP-k-accuracy as the share of articles in the test set for which the correctly linked article was part of the k top ranked candidates:

$$\text{TOP-}k\text{-Accuracy} = \frac{|\{a_\alpha \in \text{RAND1000} \mid \exists a_\beta \in \text{TOP-}k(a_\alpha) : a_\alpha \xrightarrow{ll} a_\beta\}|}{|\text{RAND1000}|}$$

One important problem in learning the classifier is the discrepancy between positive and negative training data. For every article in the training set, there exists at most one positive example — the actual target of the existing cross-language link — but up to 1000 negative examples — all other candidate articles in the restricted candidate set. Using the whole training data will most likely yield a classifier, which always predicts new examples to belong to the majority class — the negative examples in our case (compare Provost [2000]). In order to avoid this, the training data had to be balanced such that we only used a portion of the negative examples in order to train the classifier. For each article in the training set, 2, 5 and 10 negative examples were randomly selected and together with all positive examples were used to train the classifier.

To be able to measure the quality of different features, we trained the classifier with different feature sets. First, we used only the *Chain Link Count Feature*. In this case, candidate articles with a higher number of chain links are ranked higher. The

(a) Results using a ratio of 2:1 of negative/positive training example.

		TOP-k-Accuracy				
	Feature selection	*TOP-1*	*TOP-2*	*TOP-3*	*TOP-4*	*TOP-5*
1	(*Chain Link Count F.*)	42.4%	51.2%	60.0%	62.8%	64.8%
6-7	(*Text features*)	68.4%	71.2%	73.6%	74.8%	75.2%
1-5	(*Graph features*)	54.8%	64.0%	68.4%	70.8%	72.0%
1-7	(*All features*)	71.2%	76.0%	78.8%	79.6%	80.0%

(b) Results using a ratio of 5:1 of negative/positive training example.

		TOP-k-Accuracy				
	Feature selection	*TOP-1*	*TOP-2*	*TOP-3*	*TOP-4*	*TOP-5*
1	(*Chain Link Count F.*)	42.4%	51.2%	60.0%	63.2%	64.8%
6-7	(*Text features*)	68.8%	72.8%	74.4%	74.8%	75.2%
1-5	(*Graph features*)	55.2%	62.8%	67.6%	68.8%	70.0%
1-7	(*All features*)	74.8%	**79.2%**	**79.2%**	80.0%	80.4%

(c) Results using a ratio of 10:1 of negative/positive training example.

		TOP-k-Accuracy				
	Feature selection	*TOP-1*	*TOP-2*	*TOP-3*	*TOP-4*	*TOP-5*
1	(*Chain Link Count F.*)	0.0%	0.4%	0.4%	0.4%	0.4%
6-7	(*Text features*)	68.4%	72.4%	74.4%	74.8%	75.2%
1-5	(*Graph features*)	55.6%	62.4%	67.6%	69.2%	70.4%
1-7	(*All features*)	**76.0%**	78.4%	78.8%	**80.4%**	**81.2%**

Table VI.3: Results of the evaluation on the RAND1000 dataset. The tables are based on different ratios of negative/positive training examples. The first column describes the feature selection. The remaining columns show the TOP-k-accuracy of our classification approach with $k \in \{1, \ldots, 5\}$.

purpose of this experiment is to support the hypothesis that chain links are a prominent clue for language links between articles. In another set of experiments, we used the text features only as well as the graph features only, respectively. This allows to assess the influence of each of the different feature sets. Finally, the classifier was trained with all features to find out if it is indeed worth considering all the features together.

Results of the experiments are shown in Table VI.3. The table shows the accuracy with respect to the top k candidates considering varying sizes of negative examples. Overall, the choice of negative/positive ratio does not have a strong impact on the results. However, further experiments showed that using too many negative examples leads to learning a trivial classifier, as is the case when using the chain link count feature alone for a negative/positive ratio of 10:1. A negative/positive ratio of 5:1 seems therefore reasonable and will be used in the further experiments described

Ratio -/+ data	*Feature selection*	*Recall*	*Precision*
10:1	All features	69.6%	93.5%

Table VI.4: Results of the best candidate retrieval on the RANDOM1000 dataset.

below.

The accuracy of the prediction when considering only the chain link features ranges from 42.4% (TOP-1) to 64.8% (Top-5). Considering the TOP-1 results, we conclude that the classifier trained with the chain link features alone does not improve with respect to our baseline that matches articles with identical titles and has a TOP-1-accuracy of 47%. The text and graph features alone yield results in terms of accuracy between 68.8% (TOP-1) and 75.2% (TOP-5) as well as 55.2% (TOP-1) and 70% (TOP-5). Both types of features thus allow to train a classifier which outperforms the naive baseline. Considering all features yields indeed the best results, leading to a prediction accuracy between 76% (TOP-1) and 81.2% (TOP-5). Thus, we have shown that the number of chain links seems to be the weakest predictor for a cross-language link between two articles in isolation. When considering all features, the results certainly improve, showing that the number of chain links crucially contributes towards making a good decision in combination with the other features used.

As we use articles from the English and German Wikipedia as test data, the text features based on text overlap and similarity are strong features with good classification results. However, even using only graph-based features — thus operating on a completely language-independent level — the results exceed the trivial baseline. Therefore, we can assume that our method will produce reasonable results for any language pair of Wikipedia, even if they use different alphabets or if their languages are from different linguistic families. In those cases, the text based features will play a negligible role.

Best Candidate Retrieval In order to automatically generate new language links, it is necessary to choose exactly one candidate for each source article and to decide whether this candidate is the corresponding article or not. To achieve these goals, we define *best candidate retrieval* as a modified TOP-1 retrieval. Different to the TOP-k retrieval as presented above, we only select the best ranked candidate if its feature representation has a positive distance to the SVM-induced hyperplane, thus this candidate is classified to the positive class. In other cases, no cross-language links will be generated for the given source article. The results of the best candidate retrieval are presented in Table VI.4.

The recall of this experiment is 22.6% higher compared to the baseline that matches articles with identical titles. As we use the restricted candidate set that is based on the pre-selection of candidates, the maximum recall depends on the number of target articles that are actually included in the candidate sets and it has therefore

an upper bound of 86.5%. It is important to note that the recall of 69.6% of our best candidate retrieval means that we find 80% of the cross-language links that can be found at all given our pre-selection on the basis of the candidates' support.

As our aim is to learn correct links, high precision is a requirement. From this viewpoint, our approach seems very promising as new cross-language links are learned with high precision of 93.5% and a reasonable recall. It could therefore be used to enrich the Wikipedia database with new language links.

VI.3.3 Learning New Cross-language Links

In order to test our approach in a real scenario with the aim of inducing new cross-language links, instead of applying it to articles with existing ones, we processed the German Wikipedia considering all those articles which do not have an existing cross-language link to the English Wikipedia. As our algorithms are still in a state of research prototype and as we do not have the computational power, it was not possible for us to process all of these articles. Therefore, we defined a relevance ranking on the articles based on the number of incoming pagelinks and sorted the articles according to this ranking. We processed the first 12,000 articles resulting in more than 5,000 new cross-language links according to best candidate retrieval as described above.

The first 3,000 links were manually evaluated by researchers in our group. The corresponding articles of 2,198 links had identical titles. These links were assumed to be correct. The remaining 802 links were evaluated by 3 independent persons, who classified them in the following classes:

- Correct links.
- Close matches — links between articles that could be linked, but do not describe the exact same concept.
- Poor matches — links between articles that describe related concepts.
- Wrong links.

The annotator's correlation was reasonable with a Pearson's product-moment correlation coefficient between 0.80 and 0.84.

The result of the evaluation of the learned cross-language links is presented in Figure VI.3. 32% of the manually judged links are classified as correct. Adding the 8% of close matches, our approach has a precision of 40% on links that connect articles with non-identical titles.

Considering all learned links, including links that connect articles with identical titles, the overall result has a precision of 82% for learning correct cross-language links. Further, the manual evaluation showed that 92% of the links connected at least related articles. These are very satisfactory results.

This set of new learned cross-language links will be used in the next section to give an example for the self-correctiveness of Wikipedia.

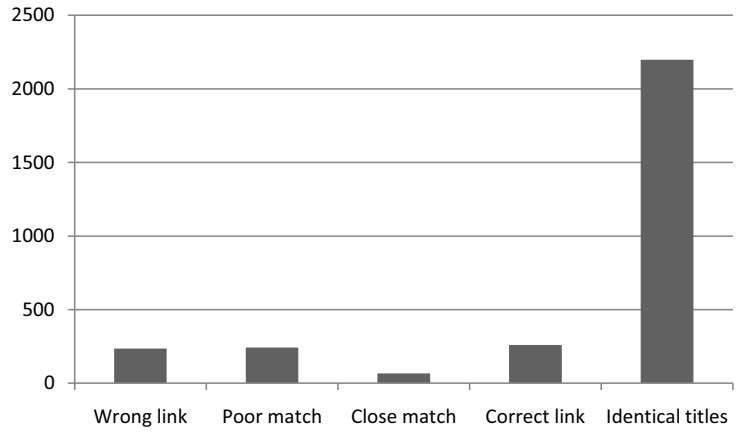


Figure VI.3: Evaluation of the first 3,000 learned cross-language links. Links connecting articles with identical titles were assumed to be correct. All other links were judged manually by three independent assessors.

VI.4 Discussion

In Chapter I, we motivated the exploitation of Web 2.0 resources presenting several advantages compared to traditional resources. Web 2.0 resources are often multilingual, cover a broad range of topics and are constantly updated. These updates include adaptation to new topics but also insertion of missing content or correction of existing errors. An interesting question is if we can prove this hypothesis and design experiments that show these constant improvements.

In this chapter, we presented an approach to identify missing cross-language links. It was applied to a dump of Wikipedia and resulted in a set of new links. We will use these links to design an experiment that shows the self-correctiveness of Wikipedia. The goal is to check the presence of these links in Wikipedia at a more recent point in time.

We also discuss the conclusions of this experiment. Our results show that Wikipedia is actually improving and structural problems are solved by the Wikipedia community. As Wikipedia is an instance of a Web 2.0 portal, conclusions can also be generalized to other Web 2.0 resources. Finally, we raise some open questions about the limitations of the self-correctiveness in Wikipedia.

VI.4.1 Self-correctiveness of Wikipedia

In our experiments, we analyze if missing cross-language links are introduced to Wikipedia over time. This intends to demonstrate the self-correctiveness of Wikipedia, in this case the introduction of missing links.

In the previous section, we identified a set of 3,000 missing cross-language links in the Wikipedia of October 2007. Our experiments are based on this set of missing links. In particular, we check if they are introduced in more recent versions of Wikipedia, in particular snapshots of September 2009 and October 2010.

The missing cross-language links were classified in the five classes presented earlier: identical titles, correct links, close matches, poor matches and wrong links. According to each class, we present statistics on the share of language links that have been added to the more recent Wikipedia versions. The results of our analysis are presented in Figure VI.4.

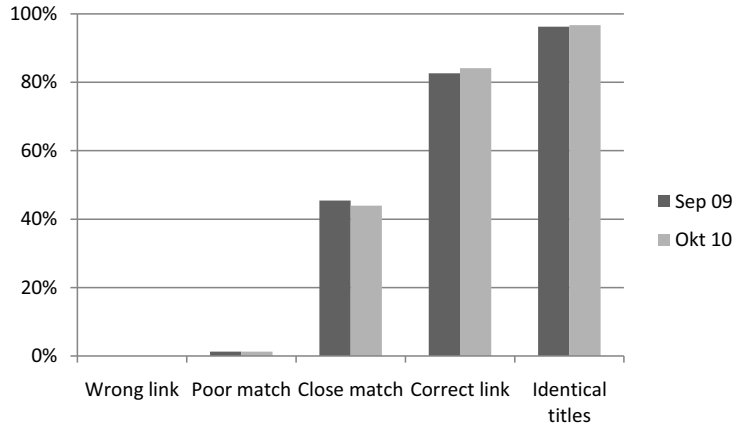
These results show that the differences between the Wikipedia of September 2009 and October 2010 are only marginal with respect to the set of missing links. All links that match to this set were added in the time period between October 2007 and September 2009. Another high level result is that there is no substantial difference between the learning rate of missing cross-language links in the English and in the German Wikipedia. Almost the same share of the 3,000 articles in the German Wikipedia have been linked to articles in the English Wikipedia as it is the case for the opposite direction.

Examining the five classes of links, the results allow the following observations:

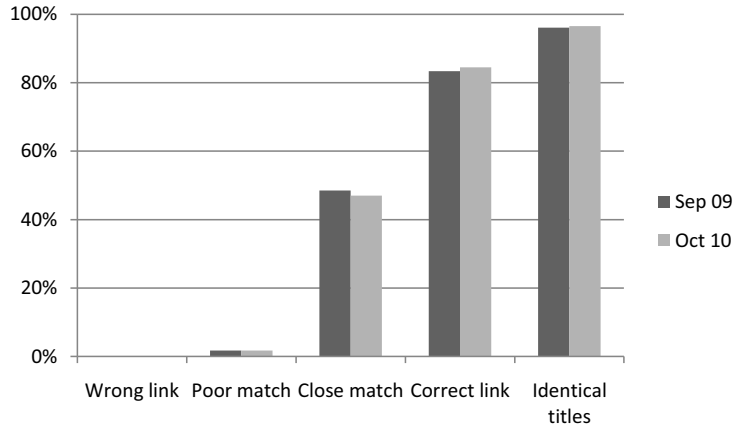
- Most of the links that connect articles with identical title have been added to Wikipedia since October 2007 — the point of time of our first experiments. In fact, more than 96% of these links have been implemented by the Wikipedia community.
- More than 83% of the links that were classified as correct have been added to Wikipedia.
- Approximately half of the links that were classified as close matches are found in the recent Wikipedia versions.
- Almost none of the poor matches and no wrong links have been added.

On the one hand, the results show that Wikipedia is able to correct deficits such as missing cross-language links. Almost all links with high confidence — links connecting articles with identical titles and links that were classified as correct — have been added to Wikipedia. On the other hand, 17% of the links classified as correct are still missing. This clearly motivates the further development of automatic approaches that support human editors. The community succeeds in finding many deficits, but will most probably not be able to find all of them.

The distribution of added language links according to the five classes as defined above also supports the classification scheme we used to evaluate the learned cross-language links. This distribution mirrors our intension when designing the target classes: links that are classified as correct should be added with high probability, links that are classified as close match might be added and all other links are probably wrong and should not be added. The data in Figure VI.4 clearly supports the outcomes of our human-based evaluation.



(a) Cross-language links from the German Wikipedia to the English Wikipedia.



(b) Cross-language links from the English Wikipedia to the German Wikipedia.

Figure VI.4: Share of the learned cross-language links present in the Wikipedia versions of September 2009 and October 2010. The links are classified based on manual evaluation.

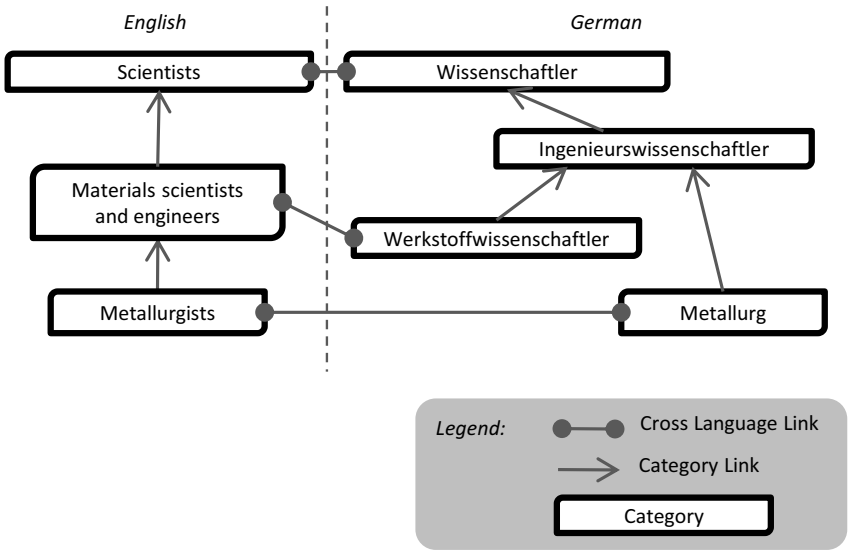


Figure VI.5: An examples for a structural difference in the category hierarchy of the English and German Wikipedia.

VI.4.2 Quality and Future of Web 2.0 Resources

Our evaluation of the missing cross-language links shows that Wikipedia, as one of the most prominent Web 2.0 resources, is constantly improving. Many Wikipedia articles are already of high quality, which is for example shown in the comparison to the Encyclopedia Britannica [Giles, 2005]. And our findings about the self-correctiveness of Wikipedia suggest that the quality of these articles is still raising. These are indications that new Web 2.0 resources replace traditional resources, which are created by a limited number of professional editors.

However, we claim that there is a limit on the self-correcting process in collaboratively created resources. One reason for this limit is the requirement of common agreement in editing content of such resources. Design decisions that are not easily judged as right or wrong will probably not be consistent based on the principles of collaborative editing. We will show this problem using the example of the cross-lingual alignment of the category hierarchy in Wikipedia.

Example VI.1 *Figure VI.5 visualizes a small excerpt of the English and German category hierarchy. In the English Wikipedia, the category* `Metallurgists` *is a sub category of* `Materials scientists and engineers` *which is a sub category of* `Scientists`*. In the German Wikipedia however, the corresponding category* `Metallurg` *(metallurgist) is not a sub category of* `Werkstoffwissenschaftler` *(materials scientist) which corresponds to*

`Materials scientists and engineers` based on cross-language links. The intermediate category `Ingenieurswissenschaftler` (engineering scientist) is instead modeled as super category of `Metallurg` (metallurgist).

The hierarchical relations between categories in the English and German Wikipedia presented in Example VI.1 illustrate the existing structural differences that depend on different viewpoints. Both versions could be regarded as correct. In a traditional encyclopedia, the consistency is enforced by strict guidelines. This is not the case for collaboratively created resources and the mechanism of common agreement will also fail to solve these structural differences. Especially when considering alignments across languages, these inconsistencies may also be grounded in linguistic or cultural differences.

When exploiting Web 2.0 resources for cross-lingual or multilingual NLP approaches, it is important to detect and to consider the inconsistencies that are introduced by content in different languages. This helps to design approaches that are robust against these structural mismatches. The overall goal is to benefit from the diversity introduced by the content in multiple languages, but also to minimize the negative influence of content that is not consistently modeled in all languages.

Apart from exploiting these resources for NLP application, the analysis of multilingual Web 2.0 resources itself is an interesting field of research. Anomalies or inconsistencies can be used to detect differences in language usage or even in culture. Understanding the multilingual interaction that is present in current Web 2.0 portals, such as Wikipedia, will be an important success factor for multilingual NLP approaches that build on this data.

Chapter VII

Conclusion

In this chapter, we first summarize the outcomes of this thesis including the main research questions and our approaches to find solutions to these questions that were presented throughout this thesis. Afterwards, we present an outlook that is based on open questions that are raised in the context of this thesis. These open questions can also be used to draft a road map for future research.

VII.1 Summary

In Chapter I, we introduced the notion of Social Semantics that describes the knowledge encoded in Web 2.0 data. We motivated the exploitation of this data in respect to multilingual IR as these collaborative created knowledges sources have several advantages compared to traditional language resources. In particular, they support multiple languages, cover a broad range of topics and almost instantly adapt to new topics.

Based on this motivation, we presented the main research questions we addressed in this thesis. In summary, these are the following questions:

(i) *Selection of Resources:* What are suitable resources that can be used as semantic background knowledge to support IR and CLIR/MLIR systems in particular, and that also meet the properties of Social Semantics?

(ii) *Application to MLIR:* How can the knowledge encoded in these resources be used to support multilingual retrieval?

(iii) *Evaluation:* What is the impact in respect to IR performance of the proposed semantic retrieval models compared to standard approaches?

Our contributions were presented in Chapters IV, V and VI. The results of our research that are described in these chapters address one or several of the research questions.

In Chapter IV, we presented a generalization and several extensions of Explicit Semantic Analysis (ESA). We used Wikipedia as a resource for conceptual knowledge that can be interpreted as Social Semantics. In respect to research question (i), we identified several outstanding properties of Wikipedia that qualify it for usage as knowledge source. This includes the definition of concepts in various languages as well as the broad coverage of topic fields. Building on the multilingual concept definitions of Wikipedia, we introduced Cross-lingual Explicit Semantic Analysis (CL-ESA), a cross-lingual extension of ESA. We also presented different design choices that allow to optimize and to adapt the CL-ESA model to specific retrieval scenarios. Concept-based retrieval based on CL-ESA is an instance of a retrieval model that exploits Social Semantics, thus addressing the research question (ii). We defined a methodology for evaluation using several standardized datasets to measure the impact of CL-ESA as demanded by research question (iii). Our results show that CL-ESA can be optimized using different parameter settings and design choices. In particular, the retrieval performance is improved if the model is adapted to the specific scenario such as CLIR or MLIR. In comparison to intrinsic concept models, we showed that CL-ESA shows comparable performance for CLIR to Latent Semantic Indexing and Latent Dirichlet Allocation. However, no training is required for CL-ESA which is a clear benefit in most application scenarios. Finally, the results of our participation at an international retrieval challenge shows that the CL-ESA-based retrieval system is competitive in CLIR scenarios.

In Chapter V, we presented an approach that is able to exploit the internal category system of a dataset in a retrieval task. In contrast to CL-ESA, no external definitions of concepts are used in the retrieval model. We propose to exploit the question and answer history and the category system in Yahoo! Answers as resources for Social Semantics, which addresses research question (i). The retrieval approach that exploits the category system of Yahoo! Answers was applied in an Expert Retrieval (ER) scenario. Our proposed retrieval model is based on a language modeling framework. A mixture language model allows to combine different sources of evidence. In respect to research question (ii), we therefore defined a retrieval model that is able to benefit from the knowledge that is provided by Social Semantics. To measure the performance of our retrieval approach as formulated in research question (iii), we refer to the topics and ground truth that we published in the context of the CriES workshop. Our results show that different sources of evidence can be successfully combined to improve the retrieval performance in the ER task. In addition, Machine Learning approaches can be used to find the best parameters for this combination. Our approaches outperform all baselines that we defined for this task as well as all runs that were submitted to the CriES challenge by other research groups.

In Chapter VI, we presented an automatic approach to improve the quality of Wikipedia. In detail, we identified missing cross-language links between existing Wikipedia articles in different languages. Our approach is based on ML and uses existing cross-language links as training data. In this context, we also analyzed the

cross-lingual link structure of Wikipedia in respect to coverage and consistency. The evaluation of the learning approach shows that we are able to identify missing cross-language links with high precision and recall. A further long term experiment that compares three Wikipedia snapshots in a time period of three years shows that more than 80% of the missing links have been added by the Wikipedia community over the time. This again proofs the quality of the learned cross-language links. Our approach to learn missing links, the analysis of existing links and the experiment about the self-improvement capabilities of Wikipedia aim at research question (i). The overall conclusion is that Wikipedia contains data of high quality which is constantly improving and can therefore be used as resource for Social Semantics.

In summary, we presented different approaches that solve many issues in the context of our three research questions. We motivated to use datasets from the Web 2.0 as resources for Social Semantics, namely Wikipedia and Yahoo! Answers. We proposed two different retrieval models that are able to exploit the Social Semantics in the retrieval process, *i.e.* concept-based retrieval exploiting Wikipedia and language models combining several sources of evidence exploiting the category system in Yahoo! Answers. Finally, we evaluated the proposed approaches using different retrieval scenarios and experimental settings on standardized datasets. Our results are promising as they show that the information encoded in collaboratively created resources can indeed be used to support NLP tasks such as IR.

In the context of the CriES workshop, we organized an IR challenge and defined a new dataset based on Yahoo! Answers, which can be used to evaluate multilingual ER systems. This exceeds our research as other research groups will benefit from the substantial contributions given by the definition of the dataset, the topics and the ground truth. Research on IR depends on these standardized retrieval settings as it allows to compare systems and to measure improvements. The CriES retrieval challenge introduces the new application area of ER in Social Question/Answer Sites that can now be used in future IR research.

We hope that our contributions advanced the research, open more research questions and will motivate future work in this field of research.

VII.2 Outlook

The outlook of this thesis covers two different aspects. Firstly, we identify some open questions that are relevant for follow up research. Secondly, we describe the possible future directions of the Web 2.0. Our definition of Social Semantics is based on datasets from the Web 2.0 and therefore its future developments are important in this field of research. We will focus on new features that will allow a more detailed and a more targeted exploitation of Web 2.0 resources.

VII.2.1 Open Questions

Open questions that can be raised in the context of this thesis are related to the *knowledge sources* and to the *retrieval systems*.

Knowledge Sources. The knowledge sources we exploit for MLIR are Web 2.0 portals that provide datasets built on the contributions of their users. There is no warranty for the quality of these datasets, and in most cases this quality will be very diverse. While some parts of the datasets may have content of high quality, other parts may be neglected. It is also very difficult to distinguish between objective and subjective content. This problem — which is analyzed in the research field of sentiment analysis — is not solved for the usually short text snippets that are submitted to Web 2.0 portals.

Another issue is the robustness of the exploitation in respect to spam or attacks. Approaches that exploit Web 2.0 datasets have to tolerate some level of spam as it will most probably not be possible to detect and to delete all contributions of spammers. The question remains how much spam in a dataset is acceptable in respect to the usefulness of the dataset as knowledge source. Other possible problems are attacks to Web 2.0 portals. These attacks aim at influencing the opinion or the perception of a specific topic. As attackers add or modify content, this can also have an affect on systems that exploit the resulting datasets. To detect spam or attacks in Web 2.0 resources and to design spam and attack tolerant systems for exploiting these resources are important issues that should be analyzed and addressed in future research.

Considering Wikipedia, the problem of spam is solved by the community — at least for Wikipedia versions such as the English or German Wikipedia which have a large number of active editors. Wikipedia is therefore a high-quality resource that can already be exploited. However, a prominent problem of Wikipedia are different opinions of editors, which could be also interpreted as attacks depending on the individual view point. In some cases, this leads to *edit wars*, *i.e.* edits are reverted and re-submitted again in a loop. Wikipedia offers tools to prevent and to end such edit wars. Still, these problems motivate to design systems that are able to deal with different opinions for the same topics, which is another direction of future research.

The privacy and accessibility of data is another problem in the Web 2.0. While this is not an issue for collaboratively created resources such as Wikipedia, it is very important for portals containing personal information such as social networks. For these resources, a balance between privacy of users and accessible information has to be found, which for example requires techniques to anonymize the data. On the one hand, all users might benefit of the publication of their data as systems that are then able to exploit this data could improve personal tasks such as IR. On the other hand, personal data has to be protected and users should always be able to control the amount of data that is publicly accessible. In many cases, it is difficult to find the balance between privacy and public benefit.

Retrieval Systems. The semantic retrieval systems that we presented in this thesis could be extended by addressing the following open research questions.

Firstly, the personalization of retrieval systems to specific users can be used to improve the retrieval performance and therefore the satisfaction of the information needs of the users. This personalization could be based on contextual information that is given by the searcher, for example connections in his/her social network. A possible approach to use this context in semantic retrieval models is to filter the background knowledge for information that is relevant for the current user and to only use this filtered knowledge in the retrieval process.

Secondly, a classification of topics could be helpful in the retrieval process. Different topics require for different background knowledge and also have different expected result types. Retrieval systems that are able to adapt to specific types of topics could therefore improve the overall retrieval results. A possible approach for this adaptation is to define a set of classes for topics and to use a classifier to assign new topics to a specific class. Based on training data, specific background knowledge for each class is identified and then used to support the retrieval for topics that are assigned to the according class. For retrieval models based on Machine Learning, specific models could be learned for each class which has the potential to improve the retrieval performance. The classification of topics could also be used to select between different retrieval models which are known to be superior for specific topic classes.

Finally, alternative ML models could be used in the retrieval process. In the models we presented in Chapter V, we trained the ML models on relevance assessments. Other alternative options are interactive search systems that are based on user feedback to train the underlying models. For these systems, techniques such as *Active Learning* could be used to minimize the level of user feedback while still achieving high learning rates.

VII.2.2 Future of the Web 2.0

In this outlook, we will also discuss some possible developments of the Web 2.0. In this thesis, we used the knowledge encoded in Web 2.0 datasets to improve IR. Therefore, we are interested in how developments of the Web 2.0 will also change the possibilities and challenges for the exploitation of Web 2.0 datasets. We will discuss these developments in respect to the application scenario of IR.

Web 2.0 and the Semantic Web. A development that has already started is the introduction of semantic annotations in the Web 2.0. This introduces the principles of the Semantic Web to user-generated content.

Annotations are added automatically, semi-automatically or manually by the users. Automatic approaches use available data to publish the corresponding semantic annotations. An examples for a source of annotations that can be added automatically are the relations between users in social networks, which are known by the

system. While the manual annotation will not be feasible for the majority of users, semi-automatic approaches are able to support the annotation process. Improved text editors are able to suggest annotations to users. Text mining approaches can be used to add semantic annotations in a further feedback loop. These approaches will spread the usage of semantic annotation and will enable users to contribute without being an expert in semantic technologies.

The advantage of having semantic annotations is that the semantics of the annotated text are well defined and can be processed automatically. When exploiting these datasets, the semantics have not to be inferred from the term distributions but can directly be used to support the target task. For example, considering the Expert Retrieval task on datasets from Social Question/Answer Site as presented in Chapter V, semantic annotations would allow to define a more detailed category structure with several dimensions based on different classification schemes. This data could probably be used to further improve the retrieval models for ER.

Other use cases of semantic annotations are linked resources and the encoding of context which will be described in the following paragraphs.

Linked Resources. Semantic annotations are often used to describe concepts or named entities. The linked data principles define a standardized approach to create unique identifiers of such concepts and named entities, *i.e.* Unique Resource Identifiers (URIs). Annotations using URIs then allow to make connections between several datasets, for example if the same concepts are annotated in more than one dataset.

In this thesis, only one dataset was exploited as background knowledge for each task. However, links between resources allow to combine several resources as background knowledge. This has the advantage that possibly more details are available for each concept. As the information is gathered across several resources, this might also add different view points in the definition of concepts. This allows to choose the particular definition that is most suitable for a specific task or context. The annotation based on URIs has also the potential to solve problems in NLP such as synonymy or disambiguation.

Context in Web 2.0. In the Web 2.0 scenario, all users not only consume but also contribute to the content of the Web. In most cases, these contributions only contain short text snippets, for example tags for images or short status updates. This makes it inherently more difficult to automatically annotate these contributions, as performance of automatic annotation approaches usually depends on the length of the text. For longer texts, annotations will be identified with higher precision and recall.

Therefore, it is important to not only consider the contributions but also the current context of users as additional information source for automatic annotation. This includes for example the current location or the current activity of a user. In a broader sense, the context also includes the social network defined by family or friends and other personal information such as education, work experience and hobbies. Much

of the information is not contained in the short text snippets that are usually submitted but has to be inferred from the context of a user. Therefore, the semantics of user contributions depend on this context and knowledge about the context is helpful for automatic annotation. Semantic technologies provide techniques that allow to encode the context so that it can be processed automatically.

Having more context published in the Web 2.0, richer information is available that motivates to develop further approaches to exploit Social Semantics. In the retrieval scenario, the context can for example be used to improve search that is tailored towards the current task or location of a user. The problem of using context for IR has for example been considered by Crestani and Ruthven [2007]. The design of IR systems that are able to exploit the semantics given by the context of users is a promising direction of future work in the research field of this thesis.

Bibliography

Barbara Abbott. The formal approach to meaning: Formal semantics and its recent developments. *Journal of Foreign Languages*, 119(1):2—20, 1999.

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining*, WSDM, pages 183—194, Palo Alto, CA, USA, 2008. ACM.

Maik Anderka and Benno Stein. The ESA retrieval model revisited. In *Proceedings of the 32nd International Conference on Research and Development in Information Retrieval*, SIGIR, page 670, Boston, MA, USA, 2009. ACM.

David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th International Conference on Machine Learning*, ICML, pages 25—32, New York, NY, USA, 2009. ACM.

Vo Ngoc Anh, Owen de Kretser, and Alistair Moffat. Vector-space ranking with effective early termination. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 35—42, New Orleans, LA, USA, 2001. ACM.

Peter Anick. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th International Conference on Research and Development in Informaion Retrieval*, SIGIR, pages 88—95, New York, NY, USA, 2003. ACM.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

Lisa Ballesteros and W. Bruce Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 84—91, Philadelphia, PA, USA, 1997. ACM.

Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. A language modeling framework for expert finding. *Information Processing & Management*, 45(1):1—19, January 2009a.

Krisztian Balog, Arjen P. de Vries, Pavel Serdyukov, Paul Thomas, and Thijs Westerveld. Overview of the trec 2009 entity track. In *TREC 2009 Working Notes*. NIST, November 2009b.

Mustapha Baziz, Mohand Boughanem, Nathalie Aussenac-Gilles, and Claude Chrisment. Semantic cores for representing documents in IR. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 1011—1017, 2005.

Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993—1022, 2003.

Stephan Bloehdorn, Roberto Basili, Marco Cammisa, and Alessandro Moschitti. Semantic kernels for text classification based on topological measures of feature similarity. In *Proceedings of the 6th International Conference on Data Mining*, ICDM, pages 808—812, 2006.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107—117, April 1998.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263—311, 1993.

C. Buckley and E. M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 25—32, Sheffield, UK, 2004. ACM.

Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13—47, 2006.

Hadumod Bußmann. *Lexikon der Sprachwissenschaft*. A. Kroner, 1983.

Xin Cao, Gao Cong, Bin Cui, Christian S. Jensen, and Ce Zhang. The use of categorization information in language models for question retrieval. In *Proceeding of the 18th Conference on Information and Knowledge Management*, CIKM, pages 265—274, Hong Kong, China, 2009. ACM.

Yunbo Cao, Jingjing Liu, Shenghua Bao, and Hang Li. Research on expert search at enterprise track of TREC 2005. In *Proceedings of 14th Text Retrieval Conference*, TREC. NIST, 2005.

Pablo Castells, Miriam Fernández, and David Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261—272, 2007.

William Cavnar and John M. Trenkle. N-gram-based text categorization. *Proceedings of the 3rd annual Symposium on Document Analysis and Information Retrieval*, pages 161—175, 1994.

Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. Modeling documents by combining semantic concepts with unsupervised statistical learning. In *Proceedings of the 7th International Conference on the Semantic Web (ISWC)*, pages 229—244, Karlsruhe, 2008. Springer.

Hsinchun Chen and Vasant Dhar. Online query refinement on information retrieval systems: a process model of searcher/system interactions. In *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 115—133, New York, NY, USA, 1990. ACM.

Philipp Cimiano, Antje Schultz, Sergej Sizov, Philipp Sorg, and Steffen Staab. Explicit versus latent concept models for cross-Language information retrieval. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI, pages 1513—1518, Pasadena, CA, USA, 2009.

Cyril Cleverdon. The cranfield tests on index language devices. *Aslib Proceedings*, 19(6):173—194, 1967.

The Unicode Consortium. *The Unicode Standard, Version 5.2.0*. The Unicode Consortium, Mountain View, CA, 2009.

Nick Craswell, Arjen P. de Vries, and Ian Soboroff. Overview of the TREC-2005 enterprise track. In *Proceedings of the 14th Text Retrieval Conference*, TREC, pages 199—205, 2005.

Fabio Crestani and Ian Ruthven. Introduction to special issue on contextual information retrieval systems. *Information Retrieval*, 10(2):111—113, 2007.

W. Bruce Croft. Combining approaches to information retrieval. In W. Bruce Croft, editor, *Advances in Information Retrieval*, volume 7, pages 1—36. Kluwer Academic Publishers, Boston, 2002.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391—407, 1990.

Susan T. Dumais, Todd A. Letsche, Michael L. Littman, and Thomas K. Landauer. Automatic cross-language retrieval using latent semantic indexing. In *Procedings of the AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, pages 15—21, 1997.

Ofer Egozi, Evgeniy Gabrilovich, and Shaul Markovitch. Concept-based feature generation and selection for information retrieval. In *Proceedings of 23rd AAAI Conference on Artificial Intelligence*, AAAI, pages 1132—1137, Chicago, IL, USA, 2008. AAAI Press.

Hui Fang. A re-examination of query expansion using lexical resources. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, ACL, pages 139—147, Columbus, Ohio, USA, 2008. ACL.

Nicola Ferro and Carol Peters. CLEF 2009 ad hoc track overview: TEL & persian tasks. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Greece, 2009.

Evgeniy Gabrilovich. *Feature generation for textual information retrieval using world knowledge*. PhD thesis, Israel Institute of Technology, Haifa, Israel, 2006.

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1606—1611, Hyderabad, India, 2007.

Jianfeng Gao, Haoliang Qi, Xinsong Xia, and Jian-Yun Nie. Linear discriminant model for information retrieval. In *Proceedings of the 28th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 290—297, Salvador, Brazil, 2005. ACM.

Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

Rakesh Gupta and Lev Ratinov. Text categorization with knowledge transfer from heterogeneous data sources. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, AAAI, pages 842—847, Chicago, IL, USA, 2008. AAAI Press.

Iryna Gurevych, Christof Müller, and Torsten Zesch. What to be? - electronic career guidance based on semantic relatedness. In *Proceedings of the 45th annual Meeting of the Association for Computational Linguistics*, ACL, pages 1032—1039, Prague, 2007. ACL.

F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A Konstan. Predictors of answer quality in online Q&A sites. In *Proceeding of the 26th annual Conference on Human Factors in Computing Systems*, CHI, pages 865—874, New York, NY, USA, 2008. ACM.

Daniel Herzig and Hristina Taneva. Multilingual expert search using linked open data as interlingual representation. In *Notebook Papers of the CLEF 2010 Labs and Workshops*, Padua, Italy, 2010.

Adrian Iftene, Bogdan Luca, Georgiana Carausu, and Madalina Merchez. Identify experts from a domain of interest. In *Notebook Papers of the CLEF 2010 Labs and Workshops*, Padua, Italy, 2010.

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 41—48, Athens, Greece, 2000. ACM.

Frederick Jelinek and Robert L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, 1980.

Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics*, ROCLING X, Taiwan, 1997.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, KDD, pages 133—142, Edmonton, Canada, 2002.

Thorsten Joachims, Bernhard Schölkopf, Christopher Burges, and Alexander Smola. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Learning*, pages 169—184. MIT Press, 1999.

John Earl Joseph. *From Whitney to Chomsky: essays in the history of American linguistics*. John Benjamins Publishing Company, 2002.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604—632, 1999.

Cody Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. *ACM Transactions on Information Systems*, 19:242—262, July 2001.

John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 111—119, New York, NY, USA, 2001. ACM.

Saskia le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 41(1):191—201, January 1992.

Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual Meeting of the Association for Computational Linguistics*, ACL, pages 25—32, College Park, MD, USA, 1999. ACL.

Johannes Leveling and Gareth J. F. Jones. HITS and misses: combining BM25 with HITS for expert search. In *Notebook Papers of the CLEF 2010 Labs and Workshops*, Padua, Italy, 2010.

V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 1966.

Sonya Liberman and Shaul Markovitch. Compact hierarchical explicit semantic representation. In *Proceedings of the Workshop on User-Contributed Knowledge and Artificial Intelligence at the International Joint Conference on Artificial Intelligence*, Pasadena, CA, USA, 2009.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, volume 1, pages 296—304, Madison, WI, USA, 1998. Morgan Kaufmann.

Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In *Cross-Language Information Retrieval*, pages 51—62. Kluwer, 1998.

David J. C. MacKay and Linda C. Bauman Peto. A hierarchical dirichlet language model. *Natural Language Engineering*, 1(03):289—308, 1995.

Rila Mandala, Takenobu Tokunaga, and Hozumi Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, SIGIR, pages 191—197, New York, NY, USA, 1999. ACM.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

Paul McNamee and James Mayfield. Character N-Gram tokenization for european language text retrieval. *Information Retrieval*, 7(1):73—97, 2004.

Saif Mohammad and Graeme Hirst. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 35—43. ACL, 2006.

Christof Müller and Iryna Gurevych. Using wikipedia and wiktionary in domain-specific information retrieval. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark, 2008.

Christof Müller and Iryna Gurevych. A study on the semantic relatedness of query and document terms in information retrieval. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1338—1347, Morristown, NJ, USA, 2009. ACL.

Christof Müller, Iryna Gurevych, and Max Mühlhäuser. Integrating semantic knowledge into text similarity and information retrieval. In *Proceedings of International Conference on Semantic Computing*, ICSC, pages 257—264, Irvine, CA, USA, 2007. IEEE.

Jian-Yun Nie. Towards a unified approach to CLIR and multilingual IR. In *Proceedings of the Cross-language Retrieval Workshop at the international Conference on Research and Development in Information Retrieval*, pages 8—14, Tampere, Finland, 2002.

Douglas Oard. A comparative study of query and document translation for cross-language information retrieval. In *Machine Translation and the Information Soup*, pages 472—483. Springer, 1998.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 562, Geneva, Italy, 2004. ACL.

Desislava Petkova and W. Bruce Croft. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, ICTAI, pages 599—608, Los Alamitos, CA, USA, 2006. IEEE.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st International Conference on Research and Development in Information Retrieval*, SIGIR, pages 275—281, Melbourne, Australia, 1998. ACM.

Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. KIM – a semantic platform for information extraction and retrieval. *Natural Language Engineering*, 10(3-4):375—392, 2004.

Martin Potthast, Benno Stein, and Maik Anderka. A Wikipedia-Based multilingual retrieval model. In *Proceedings of the 30th European Conference on Information Retrieval*, ECIR, pages 522—530, Glasgow, Scotland, 2008. Springer.

Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the Workshop on Imbalanced Data Sets at the National Conference on Artificial Intelligence*, Austin, TX, USA, 2000.

J. Ross Quinlan. *C4. 5: programs for machine learning*. Morgan Kaufmann, 1993.

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI, pages 448—453, Montreal, Quebec, Canada, 1995. Morgan Kaufmann.

Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11(95):130, 1999.

Philip Resnik and Noah A. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3):349—380, 2003.

Antonio M. Rinaldi. An ontology-driven approach for semantic information retrieval on the web. *ACM Transactions on Internet Technology*, 9(3):1—24, 2009.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 232—241, Dublin, Ireland, 1994. Springer.

Jacques Savoy. Data fusion for effective european monolingual information retrieval. In *Multilingual Information Access for Text, Speech and Images*, pages 233—244. Springer, 2005.

Georges Siolas and Florence d'Alché-Buc. Support vector machines based on a semantic kernel for text categorization. In *Proceedings of the International Joint Conference on Neural Networks*, volume 5, page 5205, Los Alamitos, CA, USA, 2000. IEEE.

Philipp Sorg and Philipp Cimiano. Cross-lingual information retrieval with explicit semantic analysis. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark, 2008a.

Philipp Sorg and Philipp Cimiano. Enriching the crosslingual link structure of Wikipedia-A classification-based approach. In *Proceedings of the Workshop on Wikipedia and Artifical Intelligence at the AAAI Conference on Artificial Intelligence*, Chicago, IL, USA, 2008b. AAAI Press.

Philipp Sorg and Philipp Cimiano. An experimental comparison of explicit semantic analysis implementations for cross-language retrieval. In *Proceedings of the 14th International Conference on Applications of Natural Language to Information Systems*, NLDB, pages 36—48, Saarbrücken, Germany, 2009. Springer.

Philipp Sorg and Philipp Cimiano. Exploiting wikipedia for cross-lingual and multilingual information retrieval. *Data and Knowledge Engineering*, submitted 2011a.

Philipp Sorg and Philipp Cimiano. Multilingual information retrieval. In *Multilingual Natural Language Processing Applications*. Pearson Education, to appear 2011b.

Philipp Sorg, Marlon Braun, David Nicolay, and Philipp Cimiano. Cross-lingual information retrieval based on multiple indexes. In *Working Notes for the CLEF 2009 Workshop*, Corfu, Greece, 2009.

Philipp Sorg, Philipp Cimiano, Antje Schultz, and Sergej Sizov. Overview of the cross-lingual expert search (CriES) pilot challenge. In *Notebook Papers of the CLEF 2010 Labs and Workshops*, Padua, Italy, 2010.

Michael Strube and Simone P. Ponzetto. WikiRelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence*, AAAI, Boston, MA, USA, 2006. AAAI Press.

Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161—197, 1998.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online QA collections. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL, pages 719—727, Columbus, OH, USA, 2008. ACL.

TNS Opinion & Social. Europeans and languages. Technical Report Special Eurobarometer 237/Wave 63.4, Report for the Eurobarometer, Brussels, 2005.

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979.

Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 61—69, Dublin, Ireland, 1994. ACM/Springer.

Denny Vrandecic, Philipp Sorg, and Rudi Studer. Language resources extracted from wikipedia. In *Proceedings of the International Conference on Knowledge Capture*, K-CAP, Banff, Canada, to appear 2011.

Ian H. Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of the Workshop on Wikipedia and Artifical Intelligence at the AAAI Conference on Artificial Intelligence*, pages 25—30, Chicago, IL, USA, 2008. AAAI Press.

S. K. M. Wong, Wojciech Ziarko, and Patrick C. N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 18—25, Montreal, Canada, 1985.

Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, SIGIR, pages 4—11, New York, NY, USA, 1996. ACM.

Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th International Conference on Information and Knowledge Management*, CIKM, pages 102—111, Arlington, VA, USA, 2006. ACM.

Torsten Zesch, Christof Müller, and Iryna Gurevych. Using wiktionary for computing semantic relatedness. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, AAAI, pages 861—866, Chicago, IL, USA, 2008. AAAI Press.

Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, CIKM, pages 403—410, Atlanta, GA, USA, 2001. ACM.

Hua-Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong-Kui Yu. Chinese lexical analysis using hierarchical hidden markov model. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, pages 63—70, Sapporo, Japan, 2003. ACL.

Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *Proceedings of the 17th International Conference on World Wide Web*, WWW, pages 715—724, New York, NY, USA, 2008. ACM.

List of Figures

List of Tables