5- unit

Part-C

Multilingual Information Retrieval:

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 7 (Cross-lingual IR) Given a collection

- **D** containing **documents** in language |p (collection language),
- **CLIR** is defined as retrieving a **ranked list of relevant documents** for a query in language l**q** (**query** language),
- with lp # lg. **D** is a monolingual collection — all documents in **D have the same language.**

Tn contrast to CLIR, Multilingual IR (MLIR) considers corpora containing documents written in different languages. It can be defined as follows: Definition 8 (Multilingual IR) Given a collection D containing documents in languages 11,...,l, with ; # 1; for 1 < i,7 < n, the task of retrieving a ranked list of relevant documents for a query in language 14. These relevant documents may thereby be distributed over all languages l1,..., tn.

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.

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.! 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?. These traditional resources are usu- ally hand crafted and cover only a limited set of topics. As these are closed systems, 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.

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

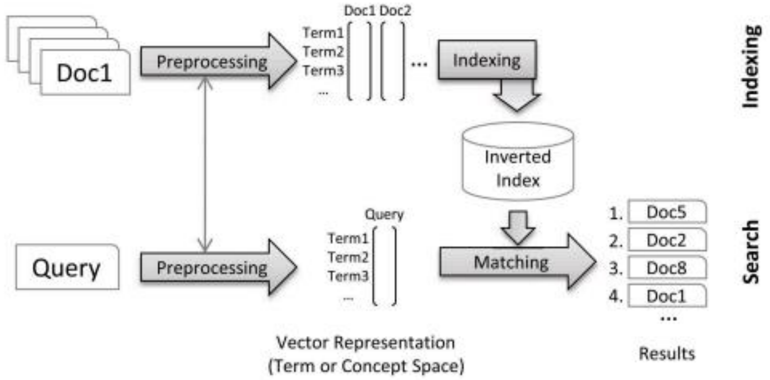


Figure II.1: The IR pipeline using vector space representation of documents and queries. The IR process consists of two processing pipelines: indexing of documents and searching of relevant documents given a query.

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.

Monolingual Information Retrieval

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

[sunni], [day], [karlsruh]

Given some simple rules, the trailing “y’” of sunny is substituted by “1” and the trailing “e” of Karlsruhe is omitted.

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 IL4. 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]. I[.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 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.

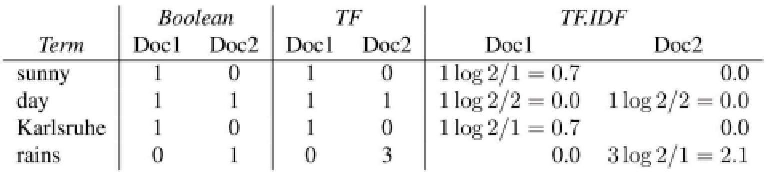
TEIDF Document Model: These models additionally multiply term frequency values by the inverse document frequency of terms. The document frequency of a term is the number of documents in the collection containing this term. The inverse document frequency therefore puts more weight on seldom terms and 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:

Docl: 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 termdocument 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	$3 \log 2/1 = 2.1$

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 —-> docl (1x)

day -> docl(1x), doc2 (1x)

Karlsruhe -> docl (1x)

rains -> docd (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].

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 realvalued 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: e 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 ¢ are based on the following measures:

— tfa(t). Term frequency in the document.

|d|. Length of the document.

df(t). Document frequency of the query term.

tfp(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:

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

Geometric Model: The vector space representation of the query q can be in- terpreted as term vector g. 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:

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

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 = Old, 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:

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

Common values for the parameters of this model are k; = 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 ¢ 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(gq) is constant for a query and P(d) can be assumed to be uniform, ranking of documents is based on the value of P(g|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 @ 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:

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

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**.**

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].

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 nonrelevant 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.

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.

Relevance Assessments

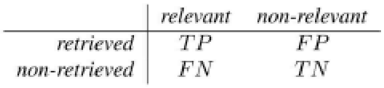
Experimentation in IR usually requires so-called relevance assessments that are used to create the gold standard. While for smaller collections, for example the original Cranfield corpus, the manual examination of all the documents for each topic by assessors is possible, this is unfeasible for larger 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 & 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.

Evaluation Measures

Tn 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 # 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.

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) [Jirvelin and Kekalainen, 2000] can then be applied as they take into account the preference to have highly-relevant documents ranked above less relevant ones.

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 crosslanguage 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. jrce.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)

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.

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* 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? 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 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)