Variations on Language Modeling for Information Retrieval

Variations on Language Modeling for Information Retrieval

Wessel Kraaij

Taaluitgeverij Neslia Paniculata / CTIT Ph.D. -thesis series No. 04-62

Graduation committee:
Prof. dr. F.M.G. de Jong, promotor
Prof. dr. ir. A.J. Mouthaan, chair/secretary
Prof. dr. T. Huibers
Prof. dr. W. Jonker
Prof. J.Y. Nie, Université de Montréal
Prof. J. Odijk, Universiteit Utrecht
Prof. dr. M. de Rijke, Universiteit van Amsterdam
Prof. K. Sparck-Jones, University of Cambridge

Taaluitgeverij Neslia Paniculata
Uitgeverij voor Lezers en Schrijvers van Talige Boeken
Nieuwe Schoolweg 28, 7514 CG Enschede, The Netherlands

CTIT Ph.D. -thesis series No. 04-62
Centre for Telematics and Information Technology
P.O. Box 217, 5700 AE Enschede, The Netherlands

CIP GEGEVENS KONINLIJKE BIBLIOTHEEK, DEN HAAG

Kraaij, Wessel

Variations on Language Modeling for Information Retrieval
W. Kraaij - Enschede: Neslia Paniculata.
Thesis Enschede - With ref. With summary
ISBN 90-75296-09-6
ISSN 1381-3617; No. 04-62 (CTIT Ph.D. -thesis series)
Subject headings: information retrieval, natural language processing

Printed by: Print Partners Ipskamp, Enschede
Cover design: Ester van de Wiel

VARIATIONS ON LANGUAGE MODELING FOR INFORMATION RETRIEVAL

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Twente, op gezag van
de rector magnificus, prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen op
vrijdag 18 juni 2004 om 13.15 uur

door

Wessel Kraaij

geboren op 14 mei 1963
te Eindhoven

Dit proefschrift is goedgekeurd door de promotor,
prof. dr. F.M.G. de Jong

in memory of my grandmother Jacoba Kraaij-Tigchelaar

to my parents

Contents

Acknowledgements

Writing the last section of this dissertation comes with feelings of great relief that this project is finally finished. It was a long project, but I think it was worth it. Despite the fact that writing this thesis implied the sacrifice of many hours of spare time, I must admit that I liked most of the endeavour, probably a *sine qua non* to complete such a big enterprise in evening hours and weekends. The most difficult part turned out to be deciding that "enough is enough" or realizing that, as my supervisor Franciska de Jong said, "better is the enemy of good". I feel very happy that I can return to a more normal life now and enjoy those activities that I had to cut down on for the last few years like spending time with friends, playing piano etc.

Three research projects provided the framework for most of the work that forms the basis for this PhD thesis. The initial plan for this thesis came up in early 1995 during the UPLIFT project on Dutch IR at the University of Utrecht. I want to thank Jan Landsbergen, initiator and supervisor of the UPLIFT project, for providing valuable feedback on the thesis material stemming from the UPLIFT period and for the suggestion to work on cross-lingual IR tasks.

After accepting a job at TNO in october 1995, I was in the fortunate position to combine some of my thesis research with several TNO projects. The work on cross-language information retrieval, which is a cornerstone of this thesis, was initially partially funded by the EU 4th framework programme, under the project Twenty-One (TAP/IE-2108) and subsequently partially funded by the Dutch Telematica Instituut under the DRUID project. Both projects were carried out in close cooperation with the University of Twente.

There are two colleagues with whom I cooperated in an intense way during these nine years and who had a significant influence on the ideas presented in this thesis. First of all, I want to thank Djoerd Hiemstra for the exciting years of participating in TREC and CLEF and for convincing me to replace the good old vector space model by language models. His focus on theoretical aspects of retrieval made me realize that techniques such as stemming, translation or query expansion cannot be treated in isolation from IR models. Secondly, I want to thank Renée Pohlmann, for many years of fruitful cooperation, resulting in several conference papers. I also want to thank her for many valuable comments that helped me to sharpen text and ideas. Both Renée and Djoerd proofread several chapters, which is greatly appreciated.

Through all these years, many colleagues at TNO were of great help for my thesis work. Rudie Ekkelenkamp was part of the TNO/UT TREC team, Giljam Derksen and Peter Defize introduced me to the controversial field of statistical significance tests and Martijn Spitters implemented the tracking system that has been used for the experiments

in chapter 7. The humour and encouragement of Stephan Raaijmakers have been a great moral support during this last year.

I feel very grateful to TNO and Jian-Yun Nie for the opportunity they offered to spend 9 months at RALI in Montréal. I had a very pleasant stay, thanks to the company of co-visitors Nikolay Vazov, Horacio Saggion and Christophe Brouard and the coffee conversations with Elliott Macklovitch and Philippe Langlais. Michel Simard and George Foster helped me to use RALI tools to create the statistical translation dictionaries, which are an important element for the work described in chapter 5.

My understanding of language modeling was considerably deepened by participating in the language modeling for IR workshop at CMU in 2001, pointed out to me by Donna Harman. The presentations of Jay Ponte, Warren Greiff, John Lafferty, Victor Lavrenko and others provided rich food for thought.

The work of David Hull has also been a great source of inspiration for me. I want to thank him in particular for his detailed comments on chapter 4 about the evaluation of IR experiments.

I really enjoyed the company and discussions with other members of the "Dutch IR clan", some of whom I met more often abroad than in the Netherlands, notably Arjen de Vries, Thijs Westerveld and Anne Diekema.

I am especially indebted to my supervisor Franciska de Jong, who stimulated me to keep on improving the manuscript while respecting a hard deadline. In spite of her busy schedule, she was always prepared to review new (and old) texts or to help with simple practical questions regarding the completion of the manuscript. Thanks to her, the period of finishing my thesis was a hectic but always pleasant time. I am honoured that Karen Sparck-Jones, Jian-Yun Nie, Theo Huibers, Maarten de Rijke, Jan Odijk and Wim Jonker accepted to participate in my graduation committee.

Several people helped me with converting the manuscript into this nice book: special thanks to Ester van de Wiel for her cover design, Hap Kolb for the LaTeX style file and Jennifer Kealy for proofreading the introductory chapter. The line patterns on the cover page are based on graphical representations of language models for each chapter of this thesis.

I would like to thank my family and friends both in Europe and North America for their encouragement and interest. Especially Khalil, Erik my brother Rik and my parents helped me to keep the thesis boat on course. My grandmother, who passed away last year at the age of 102 deserves special mentioning. She was a very lively person and always asked about the progress of my thesis work. Before I would leave on a trip abroad, she phoned me to ask whether I had made back-ups! I regret that I am not able to show her the end-result.

Finally, I would like to thank Lyne for her invaluabe and unconditional support for this project and to little Ruben I want to say: "guess what ... daddy has even more time to play with you now!"

Rotterdam, May 2004

CHAPTER 1

Introduction

The availability of a multitude of information sources via the standardized, open, network protocols of the Word Wide Web has a profound effect on society in many aspects. Information searches on the Web have become a commodity thanks to the availability of efficient search technology. Information retrieval (IR) is the area of computer science dedicated to the theory and practice of searching information. Since text is the most common medium utilized to represent and distribute information efficiently, most IR research has been focused on searches in collections of textual documents.

This thesis presents three studies in the context of search technology for text. The first two studies investigate how linguistic resources can be combined with state-of-the-art generative probabilistic IR models, also known as the language modeling approach to IR, in an effective and efficient way. In particular, we studied the use of translation resources for cross-language information retrieval and the use of different tools for morphological normalization to improve monolingual retrieval. The idea that search technology can be improved by linguistic knowledge is based on the fact that textual documents are expressions of natural language. The third study investigates whether a single document ranking model can be used for the so-called "ad hoc" retrieval task, which concerns a single retrieval session, and the topic-tracking task, which is a particular form of filtering relevant documents from a continuous stream.

The three studies can be regarded as variations on the theme of language modeling for IR. Language can either be modeled as a generative statistical process or by a collection of rules. Combining both representations of language requires special care as naive combinations may be ineffective. The title of the thesis can also be interpreted in a more narrow sense, since we also compare different configurations of statistical language models for IR tasks.

The studies are preceded by an extensive overview of state-of-the-art IR models and techniques and a study of evaluation methodologies for IR experiments. The latter is important because empirical validation is a crucial component in the development of IR systems. A general introduction to IR is given in this chapter (sections 1.1, 1.2 and 1.3). The main research questions behind the three studies are presented in section 1.4 followed by a detailed overview of the complete thesis in section 1.5.

1.1. Definition of "Information Retrieval"

The International Conference on Scientific Information held in 1958 in Washington is usually considered to be the start of IR as the field we know today (Sparck Jones, 1981). The term *Information Retrieval* (IR) was probably mentioned for the first time in a paper by Mooers (1952). It suggests a quite diverse area of R&D activities, since "information" is a fairly general term. One of the early definitions of IR by Salton (1968) indeed defines IR in a very general way:

> Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.

However, IR research as such has traditionally been focused on a particular instantiation of that task, namely retrieval of textual documents. So for a long time, Information Retrieval was more or less synonymous with *Document Retrieval* or *Text Retrieval*. More recently, new application scenarios like question answering or topic detection and tracking have become active areas of research. The recent road-map document about IR research by Allan et al. (2003) describes current IR research for a wide range of tasks. The authors note that the boundaries between the IR community and the natural language processing and database research communities are becoming less delineated since these communities have developed common areas of interest e.g., question answering, summarization and retrieval from structured documents.

Another development is that IR techniques are increasingly being adopted for non-textual material. Often so-called *multimedia retrieval* techniques are based on automatic extraction of textual or spoken parts of the multimedia documents which are subsequently processed by more or less standard text-based IR techniques. However, there is growing interest to develop media specific disclosure techniques and integrate them with well established IR-methods. The work described in this thesis is restricted to retrieval of textual documents. A more detailed description of the IR task is given in the next section.

1.2. Task description

A typical setting of an IR task involves a *user*, a *document collection* and an *IR system*. The user has a certain *information need*, which is expressed as a textual *query* and is searching for *relevant* documents in the document collection. The latter may be any kind of collection e.g., the Web or a (digital) library. The IR system must satisfy the information need of the user by analyzing both the query and the documents and presenting a list of documents to the user which are relevant to the query. This list of documents is the result of a *matching* process, that compares each document to the query. Most IR systems split the IR task in an off-line and an on-line process, in order to make retrieval on large document collections feasible:

(1) The *indexing* process, which can be carried out off-line, associates each document from the collection with an abstract representation - a *document profile* - consisting of *index terms* (often, but not necessarily equivalent to the words in the document), which characterize its content. Index terms describe a document at the content level (one of the meanings of the Latin word index is

"short description") and thus complement descriptive catalogue terms like author, title, publisher and ISBN number (Salton & McGill, 1983). The collection of abstract representations of documents described by index terms, is usually referred to as *index*.

(2) The *retrieval process*, which must be carried out on-line in an interactive setting, consists of two sub-processes. First, the user's query is analyzed and converted into a representation consisting of index terms. Subsequently, this query representation is matched with the set of index terms that represents each document and the result list is generated. The retrieval step thus consists of *query analysis*, *matching* the query representation with all document representations and *presenting* the (best) matching document references to the user.

The main function of the analysis of the query is to derive a representation which can be matched with the document representation. Therefore, queries are subjected to similar processing steps like morphological normalization. Many search engines allow a user to express the query in a certain *query language* involving e.g., Boolean or phrase operators. In this case, query analysis also includes analysis of the query structure. The semantics of these operators define extra constraints on the evaluation of the matching function.

Document profiles play an essential role in the matching process. They represent the content of documents, which is a necessity in cases where documents are not available in digital form, or access to the full document is restricted. Usually, the matching process does not access the document profiles in a sequential fashion. Instead an *index* is created to enable fast search (this process is also often referred to as indexing). The index is usually implemented as an *inverted file*: an alphabetically sorted list of index terms each of which each is paired with a list of pointers to documents. An index can be created off-line, independently of the query analysis and matching processes.

The relevance of a document with respect to a certain query is postulated to be independent of other documents. Thus retrieved documents may contain redundant information which is generally ignored in IR systems evaluation. However, there are IR performance measures which try to take this aspect into account. These measures refer to a slightly different IR task which is closer to question answering or fact retrieval, where a user simply wants to find one document which answers his question. In the standard IR setting (also referred to as the *ad hoc* task) it is assumed that a user with a certain information need is looking for as many relevant documents as possible and prefers that those documents be ranked according to relevance.

Now that we have introduced the main concepts of current information retrieval theory and practice, we will take a step back in order to show that the approach of using index terms is in fact a compromise, a pragmatic solution to the very difficult problem of interpreting and reasoning about document content and information needs.

1.3. Dealing with uncertainty

An ideal IR system should only return relevant documents, but "relevance" is very hard to formalize (Saracevic, 1975). Usually relevance is defined as a function of *aboutness*:

a document is relevant with respect to a certain information need if it is about the topics addressed in the query. More precisely, the *content* of a document is relevant to a user's need, not the document itself. It is clear that relevance and aboutness refer to the semantic content of the document and the query, but it probably also involves the task to which the user's information need is related. Since there is no accepted (open domain) knowledge representation formalism it is difficult to formalize the meaning of documents and queries. Sparck Jones and Willet formulate the IR problem as follows (Sparck Jones & Willett, 1997a):

> "The root challenge in retrieval is that user need and document content are both *unobservables*, and so is the relevance relation between them."

What is meant here is that user need, document content and relevance cannot be extracted from the surface form of the query and document by a simple algorithm. Full understanding requires a great deal of implicit contextual information, such as, information about the domain, as well as about the user's goal and pre-existing knowledge. In practical situations full knowledge of these aspects is not available. Even a more restricted content analysis procedure, which disregards context and implicit knowledge is deemed impossible due to a lack of an adequate theory of meaning and the inherent vagueness and ambiguity of language. Uncertainty with respect to meaning can thus be seen as the core problem of IR, since an IR system has to infer the information need and semantic content from the surface representations of the query and document, without an adequate theory of meaning. Moreover, the IR system has to judge whether a relevance relation pertains between query and document. The ability to handle uncertainty in an effective way seems therefore a key requirement for an IR model (van Rijsbergen, 1986).

Since a matching function based on a theory of meaning seems impossible to implement, most IR systems resort to simpler means to represent information content. One option is to use a so-called controlled language for the creation of document profiles. This has been the approach taken by library science for many centuries. The idea is to define a list (or hierarchy) of index terms with an unambiguous meaning. An example of a controlled indexing language is the Dewey decimal classification, e.g., "595.7 Insects" or "595.789 Butterflies". When documents are indexed by terms from the controlled language, and queries are composed of controlled index terms, optionally combined by Boolean operators, matching is reduced to simple lookup and set operations.

The assignment of controlled index terms to documents is clearly an intellectual process, since it involves abstraction and selection of index terms. In section 2.1, some of the main principles of manual indexing are discussed, since they illustrate some of the trade-off's that are inherent to indexing. An example of such a trade-off is the level of detail used for (manual) indexing. If a certain content aspect α of a document is not indexed, this document will never be found when a user is looking for α, which may hurt recall. On the other hand, if non-central concepts in a document are indexed, the retrieval result will be spoiled by documents which are hardly relevant, decreasing precision. Index terms thus function as an intermediate representation layer that structures a document collection; documents described by similar index terms address (at least

partially) the same topic. Adequate indexing assumes the ability to predict the terminology users will use to express their information need. High quality indexing might be attainable for trained librarians, but the task is difficult to automate due to the fact that natural language is inherently vague and ambiguous. The central role of index terms in the content representations of both documents and search statements shows that indexing and searching are tightly related: the success of a search attempt depends entirely on the quality of the indexing and query analysis procedure.

Automated methods for controlled indexing exist, based on machine learning methods or rule sets, but both approaches have important disadvantages. A disadvantage of machine learning techniques is that sufficient training data is required for each controlled term. Rule-sets are very costly to construct and maintain. In addition, all controlled indexing method require maintenance of the indexing language. Maintenance could be supported by automatic thesaurus discovery methods, but the result of these procedures often does not correspond to a human classification of the domain.

Fortunately, there is an alternative for controlled indexing, which is very well suited to automation. *Full text indexing* takes the textual representations of query and document and treats each word as an index term. This representation is also known as a *bag-of-words* representation, since all word order information is lost. Full text indexing is fundamentally different from controlled indexing, since the direct link of index terms with a (relatively) unambiguous meaning is dropped.

Basically, there are two categories of full text IR systems: *exact match* systems and *ranked retrieval* systems. The first category merely ignores the problem of uncertainty and ambiguity of index terms based on automatic full text indexing. Usually the bag-of-words is further reduced to a binary vector representing whether an index term is present of absent for a document. Many commercial full text retrieval systems use such a representation and employ exact match procedures (see also section 2.1.2.1). The advantage of this approach is its simplicity, the system retrieves only the documents which satisfy the Boolean query. However, despite its clear semantics, such an approach is not without problems, since the abstraction and selection function that were a characteristic of manual indexing are absent. The main topic of a document cannot be immediately deduced from a binary term-vector, since the Boolean model has no a-priori knowledge about which terms are more important than others.

The second category of full text retrieval systems retrieve a list of documents that are ranked to (a function of) the probability of being relevant to the users query (see also section 2.1.2.2). Such an approach supports a model representing different degrees of certainty regarding the relevance of documents with respect to a certain query. These systems try to model the importance of index terms using statistics: important terms receive a high weight and non-important terms (like function words) receive a low weight or are even discarded. Term weighting can fulfil a similar role as term selection in controlled indexing, since index terms with a very low weight will hardly contribute to the matching value between document and query. Term weighting functions can be motivated by very different modeling assumptions and are discussed at length in chapter 2. Most models do not explicitly capture meaning, but rather use the bag-of-word representation and specific model based matching functions as a means to model the relevance

relation, under the assumption that the (weighted) bag-of-word patterns implicitly encode semantic content. Of course these statistical methods cannot fully resolve the uncertainty with regard to the meaning of documents, query and relevance. Nevertheless, these methods have been proven to work in practical situations.

An important problem for IR and (knowledge representation in general) is the danger of a mismatch between the vocabulary of the user's search statement and the vocabulary used in relevant documents. This danger is not hypothetical since different groups of people often use different terms to describe the same objects or events. In order to retrieve all relevant documents, the user's query must contain those index terms that discriminate best between the relevant documents and the irrelevant documents. Documents that contain just morphological variants or synonyms of query terms are not found when relying on a basic IR model that uses full wordforms as index terms. This problem can be addressed by applying morphological normalization or using a sub-word representation. These techniques are the central theme of chapter 3.

1.4. Research questions

There are three main research questions that drive most of the work described in this thesis. The first interest is rooted in the observation that textual documents are expressions of natural language. Many researchers have tried to combine linguistic knowledge with IR systems in an attempt to improve retrieval performance. Often these approaches have been un-successful. One reason is that the combination of linguistic knowledge with IR systems has sometimes been implemented in a rather naive fashion. Most linguistic knowledge sources are compiled in dictionaries, thesauri, grammars etc. whereas IR systems model documents by weighted index terms taken from the real documents themselves. Thus it is not surprising that the linguistic resources do not boost retrieval performance significantly since they are knowledge-based while the representation format of IR systems is data-driven. These different representation types are not incompatible by definition, since experiments with a tighter integration of linguistic knowledge in the retrieval models have shown promising results. A suitable framework for an integrated modeling of query-document similarity enhanced by the use of linguistic resources is formed by generative probabilistic models of text, better known as *language models*. However, since the first publication of the application of statistical language modeling for IR in 1998, many different variants have been proposed, based on e.g., likelihood ratio, Kullback-Leibler divergence, query likelihood and document likelihood. We have studied the properties of these variations and their relationships and discuss the various alternatives in chapter 2.

The second research interest is to define a single basic but extendible formulation of language modeling for IR which is suitable for the ad hoc task, the topic tracking task and the cross-language search task. Such a definition requires a deeper understanding which aspects of the various tasks are common versus which aspects are specific. Such a single formulation is attractive from the perspective of parsimony.

IR is a good example of a field in computer science where theory and practice go hand in hand. Since experimentation is important to validate theoretically or heuristically motivated system modifications, it is also important to work with a solid methodological framework, which helps to draw conclusions that are supported by the data. The

third research question therefore focuses on the methodology used to validate experimental results and seeks to define guidelines for the evaluation of retrieval experiments. Many recommendations exist, but they are often conflicting. The guidelines have been applied for several (but not all) experiments that are reported in this thesis.

The main research questions that will be addressed in this thesis can thus be formulated as follows:

(1) How can linguistic resources be optimally embedded into IR models based on language models?
(2) Is it possible to give a single formulation of a document ranking function based on generative probabilistic models, which can be applied for various specific IR tasks: cross-language information retrieval, monolingual ad hoc retrieval and topic tracking?
(3) Is it possible to define improved guidelines for the statistical validation of IR experiments?

1.5. Thesis overview

The thesis is divided into two parts (preceded by this introductory chapter). Part I (Background) consists of three chapters. Chapter 2 gives a thorough and up-to-date survey of models for information retrieval. Indeed several introductory IR textbooks exist (Rijsbergen, 1979; Salton & McGill, 1983; Frakes & Baeza-Yates, 1992; Grossman & Frieder, 1998; Baeza-Yates & Ribeiro-Neto, 1999), but these are not always detailed enough in their explanation of the rationale behind particular term-weighting components or are limited in their treatment of different models. None of these textbooks for example discuss the application of the more recently developed language models for IR. Also, in many IR papers, authors reference a theoretical model and/or copy a term-weighting formula, but the rationale and intuitions behind the models are difficult to find or dispersed over several papers. Many of the IR systems popular among IR researchers (e.g., SMART and Okapi) have been developed over a long period, and a comprehensive overview, providing some background for this evolutionary process is not available. Chapter 2 presents some of the background knowledge required to understand the ideas behind current IR methods such as the distinction between controlled and free text indexing, or the empirical versus the model-based approach to building IR systems. Similarly, chapter 3 describes common supplementary techniques for the improvement of the performance of basic models. The chapter addresses techniques for query expansion and the application of techniques from the field of natural language processing for IR. Both aim to overcome the vocabulary mismatch problem between query and document. The final chapter of part I (chapter 4) is devoted to evaluation. In this chapter we present a review of statistical significance tests that have been applied in IR experiments. As a part of this review, we tested the assumptions of these tests on IR test data. This has lead to increased clarity regarding which methods can or cannot be applied for IR experiments. We provide explicit guidelines that describe when it makes sense to perform statistical significance tests, and which tests can be utilized. The chapters of part I were originally conceived as part of a book introducing IR to computational linguists.

Part II (Applications) describes the IR tasks that provide the context for the hypotheses that we developed in relation to the first two research questions. The hypotheses are validated by a series of experiments for each IR task. In Chapter 5 we discuss different ways to embed translation resources into a monolingual IR model based on language modeling. The resulting cross-language information retrieval (CLIR) models are evaluated in a series of contrastive experiments. Parts of this chapter have previously been published as (Kraaij et al., 2003; Kraaij & de Jong, 2004) and will also be included in a chapter in a forthcoming overview book on TREC, the annual IR evaluation conference (Hiemstra & Kraaij, 2005). In chapter 6, addressing monolingual ad hoc IR for Dutch, we discuss how linguistic intuitions about morphological normalisation in different levels of sophistication can be embedded into the basic IR model. Some of the experimental data was earlier presented in (Kraaij & Pohlmann, 1996b). Chapter 7 takes the topic tracking task as a means to investigate the behaviour of several different "language model" based IR models with regard to score normalization. This chapter is largely based on (Kraaij & Spitters, 2003). The chapters in part II can be read independently, since their topics are not inter-related. Finally, chapter 8 summarises the main results of our work and discusses them in the context of the main research questions as mentioned above and the current state of IR.

Experienced IR researchers who are interested in language models and their applications will find new variants of language modeling for several IR tasks and experimental data in part II, in particular in the section on different ways to embed translation resources in a monolingual IR model based on language models (5.2), in the section on transitive translation by matching in three different languages (5.5.4), in the section on alternative ways to incorporate morphological normalization into statistical language models (6.3.2) and in chapter 7 on score normalization of language model based ranking functions. These readers are also encouraged to look at the overview of language modeling in section 2.6.3 (the cross-entropy reduction document ranking formula that plays an important role in this thesis is presented in section 2.6.3.5). Readers interested in the application of linguistic resources for IR can find some interesting discussion and experiments in chapter 3 (overview of linguistic techniques to enhance statistical IR systems), chapter 5 (a comparison of a manually constructed and a corpus-based translation dictionary for CLIR) and chapter 6 (alternative ways to implement linguistic intuitions about morphological normalization). Chapters 2, 3 and 4 may also serve as a tutorial for entry-level PhD and graduate IR students.

PART I

Background

CHAPTER 2

Information Retrieval Models

Research in information retrieval is based on several quite different paradigms. It is important to understand the foundations of the principal approaches in order to develop a more thorough appreciation of the relative strengths and weaknesses of the different models. The history of IR research has shown that the development of models is often a combination of some theoretical modeling and a lot of experimentation guided by intuition and/or experience. This has the unfortunate result that not all of the motivations for the development of a term-weighting formula have been well-documented. In many cases, information is scattered over many different papers, sometimes with inconsistent notation. Therefore we will describe the intuitions of several important IR models in some more detail, notably the models that we have used for our IR experiments: the vector-space model, the Okapi model (Robertson & Walker, 1994) and generative probabilistic models. The chapter provides the necessary theoretical background material which serves as a starting point for our work which is presented in later chapters. It is organized as follows: section 2.1 discusses the key concepts of indexing which were developed when document retrieval was hardly automated. A lot of current problems in IR and their related terminology were already identified at that time. Section 2.2 introduces some statistical views on text and text collections because knowledge of statistics is inevitable to understand modern IR models. Sections 2.3 - 2.6 discuss the most important IR models which have been developed during the last 40 years. We will concentrate especially on probabilistic and vector space models, because these are models underlying the retrieval engines that we have used for the experiments that we describe in chapter 5,6 and 7. It is important to understand the models because one of our research questions concerns the extension of probabilistic IR models with external linguistic knowledge. The extensions can be studied in isolation, but results can usually not be generalised to a fully integrated system becuase there are usually unwanted interactions with the applied IR model. The chapter is completed with section 2.7: conclusions.

2.1. From library science to IR

Information retrieval has inherited much of its terminology from library science. The properties of indexing languages were already studied before there were automated approaches for indexing and retrieval. We will give an overview of the principal categories of indexing languages, and subsequently describe how they have been applied in manual and automated indexing situations. Principal sources for this discussion were Salton &

McGill (1983) and Sparck Jones & Willett (1997c). We will discuss the link between manual indexing and the Boolean retrieval method and contrast them with ranked retrieval systems which came into existence thanks to computers. The latter class of systems is based on the hypothesis that a list of documents ranked on relevance is the best solution to satisfy a user's information need. Because of the importance of this hypothesis for probabilistic IR models and thus for our work, we discuss it in some more detail in section 2.1.3.

2.1.1. Properties of indexing languages. There are several ways to classify the different content indexing methods. A first important distinction is whether the process is based on *controlled* versus *uncontrolled index terms*. Controlled indexing - also known as *classification* (Joyce & Needham, 1958) - limits the choice of index terms to a relatively static list which is compiled by experts. The traditional controlled indexing method has been motivated by 3 requirements (Sparck Jones, 1999):

(1) Index descriptions have to indicate the conceptual structure of a document.
(2) Index descriptions should concentrate on the source's main concepts.
(3) Index descriptions should be normalized to cope with the high variety in natural language. They should be lexically unambiguous and structurally regular.

Especially the last requirement calls for a controlled indexing language. Controlled indexing requires domain knowledge because, for example, synonym relationships have to be resolved both at indexing and retrieval time. The latter problem can be alleviated to a certain degree by adding synonyms to the list of subject headings and giving them also a separate heading with a "see: ..." reference. Controlled indexing is an activity for experts because it involves abstraction and selection, which enriches the document profile with new knowledge.

There have been attempts to automate the controlled indexing process. An early method, described in Joyce & Needham (1958) is the use of a thesaurus. In order to overcome the problem of synonymy (the user has to think about the possible terminology which could have been used in relevant documents), significant terms, for example, terms from the title or abstract, were looked up in a *thesaurus* basically consisting of headwords accompanied by a list of equivalent or closely related terms, the significant terms were subsequently replaced by the corresponding headwords. Note that the assigned index terms were taken from the thesaurus, not from the document itself. This method completely relies on the manually compiled knowledge encoded in the thesaurus. The method is restricted to a restricted domain, since it cannot cope with word sense ambiguity. Modern automatic controlled indexing systems rely on machine learning techniques. These systems learn statistical relationships between words and index terms by training on a pre-classified document collection (e.g., Masand et al., 1992; Apté et al., 1994; Schütze et al., 1995; Ng et al., 1997).

Of course controlled indexing does not prevent indexing errors. In practice, a high accuracy and consistency are hard to maintain in a group of professional indexers or in an automated system. Secondly, such a restricted *indexing language* does not give a lot of flexibility and has to be updated when, for example, a new scientific field emerges and many new concepts come into existence. Moreover a controlled indexing scheme does not allow for flexibility at retrieval time (Sparck Jones, 1999). Salton stated that

"the potential advantages of strictly controlled, manually applied indexing languages may be largely illusory" (Salton, 1989). The application of controlled indexing gradually declined in favor of an indexing approach where index terms are taken from the documents themselves. With a growing document collection, controlled indexing was found to be insufficiently discriminating.

Another distinction between indexing methods is whether the indexing vocabulary allows the use of multi-word terms or just descriptors consisting of one index term. It is obvious that an interesting query usually consists of more than one term (a single term is often too general or ambiguous). Now we could either decide to combine words to meaningful concepts at indexing time or at retrieval time. If multi-word terms are allowed in the indexing vocabulary, we speak of *precoordination*; if single index terms are combined in a query at retrieval time, this is referred to as *postcoordination*. The advantages and disadvantages of precoordinated indexing are very similar to those of manual controlled indexing. Human indexers usually assign precoordinated terms, whereas automatic indexing is usually based on single terms and coordination is only applied at retrieval time. Indexing with single terms is easy and yields reasonable results. Extension of an automatically generated index with compound terms is discussed in more detail in e.g., Strzalkowski (1995) and (Pohlmann & Kraaij, 1997a). The choice between human and automatic indexing is usually a matter of cost and quality. Human indexing has a higher quality, but is also much more expensive.

When designing an indexing method, whether pre- or postcoordinating, controlled or uncontrolled, manual or automatic, one has to consider two characteristics of the indexing method: *exhaustivity* and *specificity* (Lancaster, 1969). A document is exhaustively indexed if all concepts which are discussed are represented in the index. If a concept which is discussed in a document is not indexed, the document will not be found with a corresponding query. However, high exhaustiveness is not always desirable, since if side-issues in documents are indexed in addition to main concepts, this will deteriorate the quality of the retrieved document set. A searcher is usually not interested in documents that refer to his topic of interest as a side issue. The quality of the set of document retrieved by an IR system is usually measured in terms of *recall* and *precision*. Recall is defined as the proportion of relevant documents which is retrieved by a system, thus a high exhaustivity promotes a good recall. Precision is defined as the proportion of the retrieved documents which is relevant, thus a low exhaustivity promotes precision, since only documents which discuss the topic of interest as a main issue are retrieved. (cf. section 4.3 for a more elaborate discussion of evaluation measrures.) The specificity of an indexing language can be defined as the granularity of its index terms. If the indexing vocabulary is very specific, and each of these specific terms has a well defined meaning, it is easy to separate relevant from irrelevant documents, which increases precision of the system. On the other hand, a high specificity will cause a lower recall. For example a user interested in parrots will face a low precision when the indexing vocabulary only contains "birds", and a user interested in documents about birds might possibly miss relevant documents in a very specific indexing language since he has to enumerate all bird species in his query. Specificity is thus inversely related to the level of abstraction. For both specificity and exhaustiveness there is a trade-off between recall and precision, the optimum levels depend on the specific user population. Usually high precision is

preferred over high recall, but in specific cases (e.g., legal or patent search) high recall is important. In the manual indexing case the level of exhaustiveness and specificity are directly related to the amount of manual labour to be performed at indexing time. If high exhaustiveness and or specificity is required this might not be feasible for economic reasons. However, an alternative exists in the form of automatic post-coordinative indexing approaches. Here the exhaustivity easily reaches a higher level, because all content terms are used as index terms. A basic automatic approach however, lacks any abstraction or recognition of compound terminology, post-coordination of query terms can compensate for this to a certain extent.

The core problem of IR is thus to define optimal representation schemes for documents and information needs and to devise a complementary optimal matching function. Summarizing the discussion, we can enumerate the following desired properties for such a representation scheme:

(1) The representation scheme must allow searches with a high precision. A low precision will in fact discourage a user to keep on using the system.
(2) The representation scheme must be able to cope with terminology mismatches between the query and relevant documents. Terminology mismatches are the major cause for the low recall of IR systems.
(3) The representation scheme must be easy to manage. If we are considering manual indexing this means that it must be easy to find near duplicates, remove, add or merge indexing terms. In the case of automatic indexing, scalability is an issue. E.g., taking every maximal noun phrase[1] in a document as an index term will produce an extremely large index which is difficult to manage, because inversion is a resource consuming process. Moreover most of these maximal noun phrases are too specific.
(4) Document representations must be produced in a cost effective way.
(5) Document representations must only cover the major content aspects. Index descriptions are essentially reductive (Sparck Jones, 1992), because not everything in a text is important. Manual indexing thus always involves some kind of selection.
(6) The combination of document representations and the query language must allow an effective separation between relevant and non-relevant documents (specificity). If index terms are too general, it is impossible to separate out marginally relevant documents from documents of high relevance. On the other hand, an indexing language with high specificity should provide mechanisms to enhance recall for more generic questions.

Classical library retrieval methods are exclusively based on *exact match retrieval* models. Documents are represented by a set of *index terms*, sometimes called *keywords*. The interpretation of each index term is that the document is *about* the concept described by that index term. In the case of a pre-coordinated system, queries consist of a single index term, and the matching function will return those documents which contain the query term in their profile. In the case of a post-coordinated system, a query is represented

[1] A maximal noun phrase is a complex constituent consisting of a base noun phrase and several modifiers, e.g., prepositional phrases etc.)

by a Boolean combination of index terms. Of course evaluation of a post-coordinated query on a document base of an interesting size is only possible when the Boolean expression can be evaluated in an automated way (cf. section 2.1.2). In both pre- and post-coordinated systems, retrieval and matching procedures do not directly inspect the document profiles. This process would be too slow. Instead they access the index which has been produced from the document profiles.

Designing an indexing language which meets all desired properties is quite difficult, because the requirements conflict with each other. Suppose we want to enable high precision searches, then it is favourable to index documents with quite detailed multi-word terms. However, such an approach will affect the second property, because the more detailed the index terms are, the more difficult it will be for a user to create a query that will retrieval all relevant documents. If the matching function would be based on an exact match, probably very many relevant index terms would be missed; this problem can be alleviated to some extent by browsing the list of terms. But browsing an index term list for index term selection is certainly not a scalable solution when the index terms are quite detailed. Assigning long, precise index terms leads to a combinatorial explosion when we increase the number of documents.

The same trade-off which is apparent between recall and precision applies to the last three properties. If we want to cover the major content aspects of a publication with a high specificity, this will cost considerably more effort than only assigning a term for the main theme. But note that automatic methods can help here. We will give an overview of the basic terminology and concepts of automatic indexing in the next subsection. A much more elaborate treatment of statistical IR models follows in section 2.3.

2.1.2. Introduction into automatic indexing. The introduction of the computer for document retrieval purposes marks the start of IR as a separate field in computer science. The computer can be applied for any IR approach, be it pre- or post-coordinative, with controlled or uncontrolled index terms. However, most automated IR systems are based on post-coordinated uncontrolled indexing terms. At first these methods were applied to (manually generated) abstracts but later when documents became available in electronic version, *full-text indexing*[2] became common practice. The automated post-coordinated uncontrolled indexing approach to generate document profiles is often referred to as *bag of words* indexing, since all words in the documents (filtered through a so-called stop list) are included as index terms.

2.1.2.1. *Exact match retrieval.* Most early elementary automated IR systems were (and still are) based on the Boolean retrieval model; this is not surprising since the Boolean model has been preferred by search professionals and naive WWW searchers, because of the clear semantics of the matching function. Only documents which satisfy the query (which is formulated as a Boolean proposition) will be returned. For this reason such systems are also called exact match systems. A Boolean system in combination with a controlled indexing language can be an effective tool for professional librarians. The Boolean model is less effective when applied to automatic uncontrolled full-text indexing. An important assumption of Boolean retrieval models is that when a document profile contains a certain index term, it is assumed that the document is *about* this index term,

[2]Sometimes the term *free text* indexing is used

since the Boolean approach does not model uncertainty and lacks term weighting. Automated full text indexing approaches for Boolean retrieval treat every content word of the document as an index term. It is easy to show that the aboutness assumption does not hold in general for uncontrolled (free) index terms: a document which contains the word *world* is probably not about the earth. A possible remedy is to use (post)-coordination. Documents that satisfy the query *third AND world* have a large probability that they discuss some aspect related to third-world countries. However, the more conjuncts we add to the query in order to enhance precision, the larger the probability is that the system will return no documents, so full conjunctive queries are less useful for longer queries. The Boolean model and its variants will be discussed in more detail in section 2.4.

2.1.2.2. *Ranked retrieval.* Another option to deal with the violation of the aboutness assumption is to try to rank the index terms. Usually such a ranking is based on both global statistics from the document collection and local frequency counts in the document. These statistics help to capture two intuitions:

(1) An index term which occurs in a lot of documents is not very discriminative, therefore the weight of an index term should be inversely proportional to the number of documents in which it occurs
(2) An index term which occurs very often in a document is probably highly relevant for that document, therefore the weight of an index term should be proportional to the number of occurrences within the document

Relevance ranked systems differ from Boolean systems in two principal aspects: (i) Boolean systems start from the aboutness assumption, whereas relevance ranked systems accept that occurrence of an index term is an uncertain indicator for aboutness (ii) Boolean systems retrieve a set of documents with no internal ordering whereas ranked retrieval systems retrieve a list of documents sorted by their (estimated) relevance[3]. Because the relevance of a document given a certain query can only be estimated, IR systems differ fundamentally from database systems, which retrieve documents that satisfy certain constraints. These constraints can be simply evaluated by checking the attributes of each object in the database, no uncertain inference is involved. Systems based on relevance ranking will be discussed in more detail in sections 2.5 - 2.6.

2.1.2.3. *Basic indexing process.* The basic processes to derive a content representation suitable for post-coordinative retrieval models (e.g., Boolean or relevance ranked models) from a (full) text involve the following steps:

- *tokenization*: converting a full text into a list of tokens which define the content of the text, this involves deleting markup codes, characterset normalization etc. This thesis will not discuss tokenization.
- *term selection*: deciding which of the tokens are relevant for a content description of the document. This process usually involves at least removing *stopwords*. A so-called *stopword list* usually consists of function words, sometimes complemented with some high frequency words. Section 3.4 discusses stop lists in more detail.

[3]Often systems do not compute the absolute probability of relevance of a document but a derived document score which preserves the relative ordering of documents.

- *term normalization*: In order to remove the redundancy which is caused by morphological variants, terms are normalized to a canonical form, a typical example is *stemming* (cf. chapter 6).
- *term weighting* (only for ranked retrieval systems): Since the limits of pure Boolean retrieval models were already discovered quite early, several proposals for effective query term weighting have been developed. Sections 2.3- 2.6 give an elaborate overview of term weighting models.

In principle, these processing steps are also applied to the free format search statement that expresses the user's information need.

Automatic IR systems can also be based on different indexing units. For example, for dealing with OCR'ed input data the use of character n-grams has been investigated (de Heer, 1979; Mittendorf, 1998), A good overview of different approaches can be found in the report on the TREC-5[4] confusion track (Kantor & Voorhees, 1997). Some researchers have investigated the use of character n-grams as indexing units on normal, un-corrupted, text collections. The use of n-grams can have two potential advantages: (i) it provides a kind of approximate string matching, potentially improving recall, (ii) some phrasal structure is encoded in the index descriptions when when overlapping n-grams are used (which span word boundaries). The results of early experiments using 4-grams by Cavnar (1995) are hard to assess because no word-baseline results were reported. A more recent experiment compared an approach on full (un-stemmed) words with word boundary overlapping 5-grams (Mayfield & McNamee, 1999). The 5-gram approach performed significantly better. Unfortunately no comparison experiment with stemmed words was done. Finally, also hybrid approaches where a document is indexed both by words and by other index descriptions like n-grams have been investigated by Mayfield & McNamee (1999). Usually such *fusion* approaches improve results. Cf. section 3.2.2 for some further discussion of the use of n-grams for indexing.

2.1.3. Probability Ranking Principle. In fact it was already 40 years ago that the first proposal for a probabilistic indexing method was put forward in (Maron & Kuhns, 1960). The key contribution of this paper is that index terms are uncertain predictors of relevance. Maron and Kuhns proposed to weight the query terms on the basis of the term distributions in the documents to arrive at a statistical inference scheme which allows the computation of a notion of relevance which is suitable for relative comparisons of relevance. The authors simply a-priori accepted the principle that ranking the documents with respect to their relative relevance is optimal:

> "Finally, the paper suggests an interpretation of the whole library problem as one where the request is considered as a clue on the basis of which the library system makes a concatenated statistical inference in order to provide as an output an ordered list of those documents which most probably satisfy the information needs of the user."

Because a retrieval system cannot predict with certainty whether a certain document will be relevant to a user's information need, it will necessarily be probabilistic in nature. Systems will have to predict the probability of relevance of a document given the available data. This initial formulation of the probability ranking principle has been amended

[4]TREC (=Text REtrieval Conference) is the most important IR system evaluation activity, cf. section 4.2.

and criticized by several authors, notably Cooper and Robertson. Cooper gave a new formulation of the PRP (Cooper, 1971, 1994)

> "HYPOTHESIS: If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data."

A formal justification of the PRP has been given in 1977 (Robertson, 1977). For the restricted situation of one user with one request one can justify that the PRP leads to an optimal IR system. We will replicate one of the justifications below. Robertson presents the proofs given a dichotomous relevance notion (a document is either relevant or not relevant to a user's request). Of course this is quite a crude assumption, but as several authors have shown (cf. chapter 4) a more refined relevance variable (for example a continuous variable) does not lead to better IR models/systems. Such a dichotomous relevance variable enables the definition of the probability of relevance. Another important assumption is that the relevance of a document is independent of the relevance of other documents. We will replicate the second formal justification of PRP from (Robertson, 1977) which is a decision theoretic argument.[5]

Suppose a system must decide whether to present the document to the user or not given a certain probability of relevance computed by the system. Two "cost functions" can be defined which describe the cost of making an erronic decision:

$$Cost(\text{retrieved}|\text{not relevant}) = c_1$$

$$Cost(\text{not retrieved}|\text{relevant}) = c_2$$

It is assumed that the system has some estimate about the probability of relevance $\phi(d_i)$ of a document d_i. The expected cost to retrieve a document can be computed as follows:

$$(1 - \phi(d_i))c_1$$

or not to retrieve it:

$$\phi(d_i)c_2$$

The total cost of retrieving documents can be minimized when documents are only retrieved if:

$$\phi(d_i))c_2 > (1 - \phi(d_i))c_1$$

or:

$$\phi(d_i) > \frac{c_1}{c_1 + c_2}$$

This implies that the optimum ranking of the hypothetical probabilistic IR system is to rank in $\phi(d_i)$ order, and stop retrieving documents when the threshold $c_1/(c_1 + c_2)$ is reached.

---

[5]Robertson also gives a justification in probabilistic terms, based on the binary independence retrieval model which is explained in section 2.6.1).

Summarizing, we can state that the observation that index terms are uncertain predictors of relevance has lead to a very influential class of IR models which are based on the PRP. In contrast to Boolean retrieval systems, these statistical systems try to model a form of relative relevance, either by probabilistic means or by defining a similarity function in a high dimensional space (cf. section 2.5). The intuitive idea that documents have to be ordered to their probability of relevance can be explained by a decision theoretic argument.

2.2. Statistical properties of text

Because most IR models are of a statistic nature, they will either explicitly or implicitly assume a certain distribution of the textual data. Assuming that the data has certain statistical properties makes it possible to draw statistical inferences. Well known distributions from statistics are the normal or Gaussian distribution and the binomial distribution. The former is a *continuous* distribution, where the random variable has a continuous domain, the latter is a *discrete* distribution, in this case with two possible values for the random variable. We refer to Manning & Schütze (1999) for a solid overview of distributions which are used to model textual data in general and linguistic phenomena like phrase structure or collocations in particular.

In the following subsections we will discuss some of the distributions which have been used to model textual data in the context of IR.

2.2.1. Zipf's laws. Some early studies on the distribution of words were carried out by George Zipf (Zipf, 1949). Zipf studied language use in a text corpus and found several empirical laws which he presented as empirical evidence for his *Principle of Least Effort*. One of these principles is often quoted as Zipf's law[6], it describes the distribution of word frequencies in a corpus. When we make a histogram of words occurring in a text corpus and sort the words to descending frequency, we see a non linear curve. The distribution is not homogeneous but *skewed*. Zipf approximated the shape of this histogram with the formula

$$\text{frequency} \times \text{rank} = \text{constant} \tag{1}$$

This hyperbolic curve (see Fig. 2.1 for an example) reflects the fact that there is a small vocabulary which accounts for a large part of the *tokens* in text. These words are mainly function words. Manning & Schütze (1999) did an analysis on the novel *Tom Sawyer*, containing 11.000 word *types*. There are twelve words (*the, and, a* etc.) which each account for more than 1 % of the tokens in the novel. On the other hand, the types that occur only three times or less account also for 12 % of the total number of tokens, but this time the number of types is roughly 8550. "Zipf's law" has been tested on several corpora and has shown to be a good first order approximation; better approximations exist (the Mandelbrot distribution) but are not relevant for our study. Zipf explains the hyperbolic distribution by what he calls the least effort principle, assuming that it is easier for a writer to repeat certain words instead of using new words. (A listener however, would prefer different more infrequent words with an unambiguous meaning.) There is however a much simpler and more quantitative model for the rank-frequency

[6]Cf. http://linkage.rockefeller.edu/wli/zipf/ for a complete overview of relevant literature.

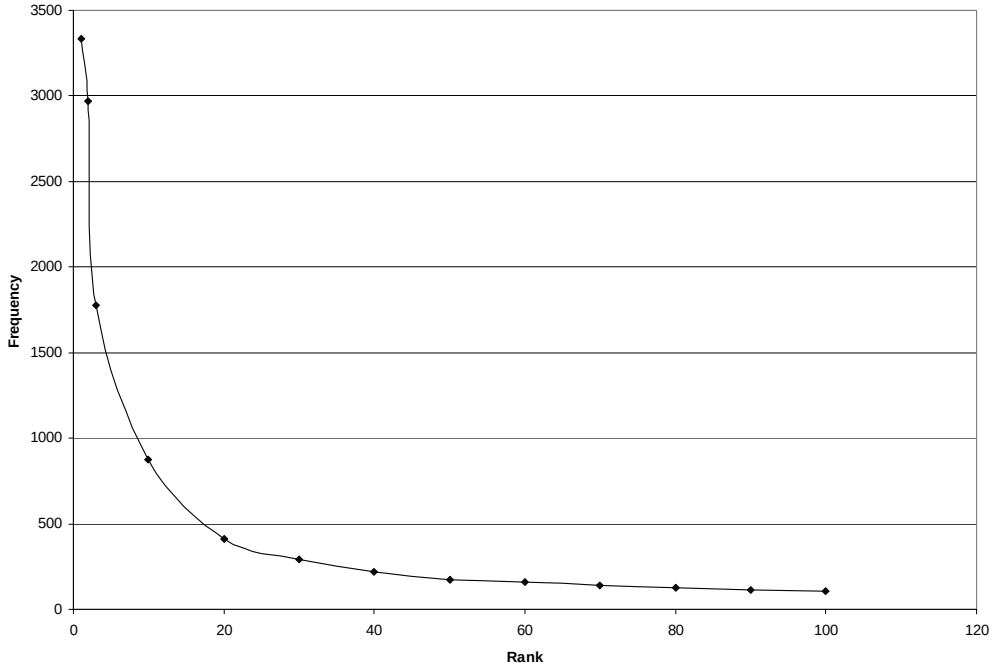

Figure 2.1. Type frequencies, sorted in descending order

law. Suppose we generate a text (the original example in (Miller et al., 1957) starred typing monkeys) with a probability p to generate a space and a probability $(1 - p)$ to type a letter, each letter having an equal probability. Then the result will model the Zipf distribution quite well. The accuracy of this model (the match to actual empirical data) is even further improved when the probabilities of the individual letters are not uniformly distributed but are estimated on a large text corpus (Bell et al., 1990). This match does not prove Zipf's "law", but it shows that it can be explained by a simple generative model. The Zipf distribution shows that the major part of the types in a text are quite rare, which poses practical problems for parameter estimation in statistical IR models: the *sparse data problem*. On the other hand, the reciprocal relationship between rank and frequency could be taken as a starting point for index term selection (Salton, 1989). The idea is to sort word types according to their frequency in a corpus. As a second step, the high frequency words can be eliminated because they do not discriminate well between the documents in the collection. Thirdly low frequency terms below a certain threshold (e.g words that occur just once or twice) can be removed because they occur so infrequently that they are seldom used in user's queries. Using this approach, the size of an index can be reduced significantly. A more principled approach to differentation between index terms is term weighting. In term weighting models, mid frequency terms turn out to be the most important indexing terms, since they discriminate well but are not to specific. Several models for term weighting will be discussed in section 2.5 and 2.6.

Another empirical law postulated by Zipf is that the number of meanings of a word is correlated with the square root of its frequency. This would imply that infrequent words are less ambiguous, and would confirm that high frequency words are not suitable for index terms. Zipf also found that the length of a word is inversely related to its frequency, which can easily be verified by inspecting a list of function words. The latter law indeed serves as an example of the principle of economy: shorter words require less effort and are thus coined more frequently. We can also explain this 'law' by looking at the generative model. It is easy to see that the probability of a word decreases with its length, the probability of generating n non-spaces terminated with a space is: $(1-p)^n \cdot p$, where p is the probability of generating a space.

Though Zipf's law gives interesting general characterizations of words in a corpus, it is not useful for the statistical characterization of distinct documents in a *text collection* i.e. a corpus which consists of distinct independent documents. IR systems try to order documents according to their relative probability of relevance given a certain query (cf. PRP section 2.1.3). For the estimation of this probability we will need a characterization of the occurrence or frequency distribution of the query terms in individual documents in relation to their global distribution.

Recently it was shown that word senses also have a skewed frequency distribution (Sanderson & van Rijsbergen, 1999). Because one sense of an ambiguous word accounts for the major part of its occurrences, IR systems are relatively robust to word sense ambiguity. The problem of word sense ambiguity will be further discussed in section 3.3.3.

2.2.2. The binomial distribution. The binomial distribution is one of the standard statistical distributions. It concerns the outcome of a series of (independent) *Bernoulli trials*, e.g., a random event with two possible outcomes, like flipping a coin. The number of heads r after n trials exhibits a binomial distribution given a probability for a head p is

$$b(r;n,p) = \binom{n}{r} p^r (1-p)^{n-r} = \frac{n!}{(n-r)!r!} \quad p^r (1-p)^{n-r}$$

For a perfect coin, $p = 0.5$. One could choose the binomial distribution to model the occurrence of a word in a text corpus. A text corpus is in this case seen as a sequence of n trials, where p represents the probability that a word occurs, and $(1-p)$ the probability that another word is generated. A nice property of the binomial distribution is that it can be approximated by the normal distribution when $np(1-p) > 5$. This makes the full repertoire of statistical instruments available for cases where n is large and p is not too small. One could for instance use a t-test[7] to investigate whether the number of occurrences of a word in a document is *significantly* higher than what could be expected on the basis of global, collection wide, word counts. High significance could be a good indicator for a good index term.

Unfortunately, in text analysis the assumptions required for normality approximation often do not hold because of the sparse data problem: We know from Zipf's law that most words occur very rarely. For example, the UPLIFT[8] corpus contains 26,719,503

[7] A statistical hypothesis test compares observed data with the distribution which is hypothesized and yields the probability that this hypothesis is true. Cf. chapter 4 for a more elaborate discussion of hypothesis testing.

[8] The UPLIFT corpus is a Dutch IR test collection by Kraaij&Pohlmann. See chapter 4.

tokens and 433,226 types in 59,608 documents. So the average document length is 129. For a term that occurs 100 times in the corpus, this would give a np of 0.00048, which is far below the thresholds for approximation with a normal distribution. This problem is even worse for phrases. This implies that the t-test is invalid for most infrequent words in a corpus and especially for phrases. An alternative for the tests based on the normality assumption are likelihood ratios based on binomial or multinomial distributions (Dunning, 1993).

2.2.3. The Multinomial distribution. The multinomial distribution is an extension of the binomial distribution. We assume a discrete sample space, where a trial can have m outcomes (instead of two in the binomial case). We can model the probability that each of the m outcomes occurs with a frequency f_i after n trials:

$$m(f_1, f_2, f_3, \ldots f_m; n, p_1, p_2, p_3, \ldots p_m) = \frac{n!}{f_1! f_2! f_3! \ldots f_m!} \quad {p_1}^{f_1} {p_2}^{f_3} p_3 \ldots {p_m}^{f_m} \tag{2}$$

where $\sum_{i=1}^{m} p_i = 1$ and $\sum_{i=1}^{m} f_i = n$. (2) can be reformulated as (3):

$$m(S) = \frac{n!}{\prod_{t=1}^{m} f_t!} \prod_{t=1}^{m} p_t^{f_t} \tag{3}$$

where $m(S)$ denotes the probability that the sentence S is drawn from a multinomial distribution.

The probability of a certain sequence of events[9] (assuming that the events are independent) can be modeled by the multiplication of the probabilities of the individual events:

$$P(T_1, T_2, \ldots, T_n) = \prod_{i=1}^{n} P(T_i) \tag{4}$$

An example of a multinomial distribution is a word unigram model, which corresponds to a zeroth order Markov Model, without any state history. The multinomial distribution is applied by several researchers for IR purposes (cf. section 2.6.3). The intuition here is that the probability of relevance of a document with respect to a query can be modeled by the probability that the query is generated by a unigram model of which the parameters are estimated from the document. In other words: for each document we build a small statistical language model and estimate the probability that it generated the query (e.g., Hiemstra, 1998).

2.2.4. The Poisson distribution. The Poisson distribution is one of the standard probabilistic distributions which is used to model the number of occurrences of a certain random event in fixed size samples, e.g., the number of typos which are produced on a page. The Poisson distribution is described by

$$p(k; \lambda_i) = e^{\lambda_i} \lambda_i^k / k!$$

where $p(k; \lambda_i)$ is the probability that a certain event i occurs k times in a unit. The Poisson distribution has the interesting property that both expectation and variance are equal to λ_i. The Poisson distribution is a limit of the binomial distribution where the

[9] Note that we model ordered events here.

number of trials approaches to infinity and the probability p is approaching zero, while $n.p$ remains equal to λ_i.

The Poisson distribution has been used in IR to model the distribution of terms over documents, i.e. we apply the model to predict the probability of the term frequency k of a certain term i in a random document: $P_i(k) = p(k;\lambda_i)$. The parameter λ_i is the average term frequency of term i in the collection which is equal to the global term frequency gtf_i[10] (number of occurrences of term i in the collection) divided by the number of documents.

The Poisson distribution makes some assumptions which do not hold for actual text data.

(1) *The probability of more than one occurrence of a term is much smaller than the probability of one occurrence.* In reality, when a term is used, it is often used more than once (*burstiness*). In reality, terms are not independent, which is an assumption of Poisson. The deviation between predicted and observed frequency is especially prominent for content terms[11], which are of prime importance for IR.

(2) *Poisson models the frequency of occurrence in a fixed interval.* In reality however, the length of documents in a collection is extremely variable, length differences of a factor of 100 or more do occur quite frequently.

The simple fact that one assumption does not hold, does not always invalidate the approach (as we shall see again in chapter 4). But in this case the deviations between observed and predicted data are so large that more refined models have been proposed. However, more complex models do not always solve the problem; they impose a larger computational complexity which is not desirable for today's large scale applications.

2.2.5. The 2-Poisson model. A model which provides a better fit of the term frequency distribution of content terms is the *2-Poisson Model* (Bookstein & Swanson, 1975; Harter, 1975). It is assumed that a collection of documents can be divided in two classes, a document is either about a certain term (*elite*) or not (non-elite). Both document classes are modeled by a Poisson distribution, but the probability of a term i occurring k times is in this case modeled by combining the estimates from both models:

$$2p(k;\lambda_1,\lambda_2) = \alpha e^{\lambda_1}\frac{\lambda_1^k}{k!} + (1-\alpha)e^{\lambda_2}\frac{\lambda_2^k}{k!} \tag{5}$$

where λ_1 and λ_2 are the average number of occurrences in the class of elite and non-elite documents respectively, α is the probability that a document is relevant. The 2-Poisson model postulates that a word can either be of central importance for the content of a document, or occurs spuriously and should not be considered as an index term. The technique of using mixture distributions plays is also applied in more recent statistical approaches to IR, which will be discussed in section 2.6.3. Section 2.6.1.2 discusses the Okapi IR model which is based on the 2-Poisson model.

[10] Manning & Schütze (1999) use the term collection frequency, which has a different meaning in IR.

[11] Manning & Schütze (1999) mention an overestimation factor of the estimator 3 to 4 times the real parameter value.

2.3. Overview of IR Models

In this section we introduce the main approaches to IR, which will be elaborated upon in sections 2.4, 2.5 and 2.6. We will start by recapitulating the basic notions in IR in a conceptual way.

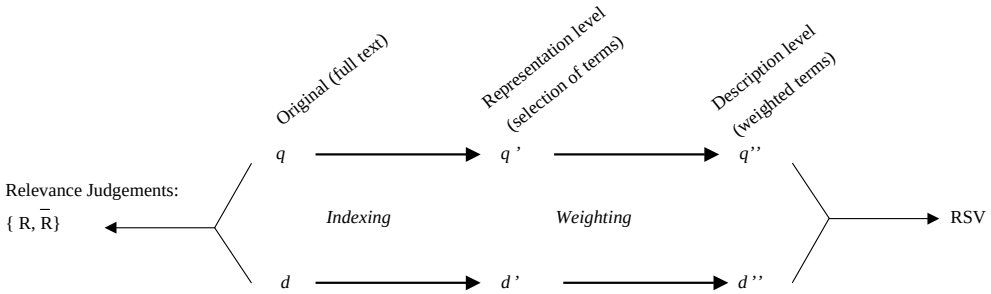

Figure 2.2. Conceptual scheme of IR, after (Fuhr, 1992). d = document, q = query.

2.3.1. Conceptual scheme of the IR process. Figure 2.2 presents a conceptual schema of the IR processing steps. The IR task consists of a user that poses a certain query q, a collection of documents $d_1, d_2, ..., d_N$ and an IR system. The *indexing* process consists basically of term selection, because conventional automated IR systems work with full text documents in a post-coordinated retrieval setting. The indexing process thus extracts the representations q' for the query and d' for each document. This representation level is used by the classical Boolean retrieval model, more advanced IR models apply term *weighting*, yielding the descriptions q'' and d''. The IR system finally applies a matching function $R(q_i, d_j)$ which computes a ranking score (retrieval status value: RSV) for each document d_j given a query q_i. Apart from the query and document descriptions, the ranking function usually uses global collection statistics. Finally the results of the retrieval process can be evaluated by judging the relevance relation between the document and the query. Usually, in IR evaluation, relevance is taken to be dichotomous: a document is either relevant (R) or not relevant ($\bar{R}$). Chapter 4 will discuss the evaluation process in more detail.

2.3.2. Taxonomy of IR models. The three main classes or IR models are:

- logical models[12]
- vector space models
- probabilistic models

It is hard to devise a reference taxonomy of all known IR models, because there are different views to classify the different models. Figure 2.3 presents the classification that we will use to present our survey. The structure is reflected in the section numbering. To simplify the picture (we would have needed more dimensions) we left out two further classifications: (i) relevance ranking vs. exact match: the Boolean model is the only

[12]For presentation purposes, we classify both Boolean retrieval and models based on non-classical logics as logical approaches.

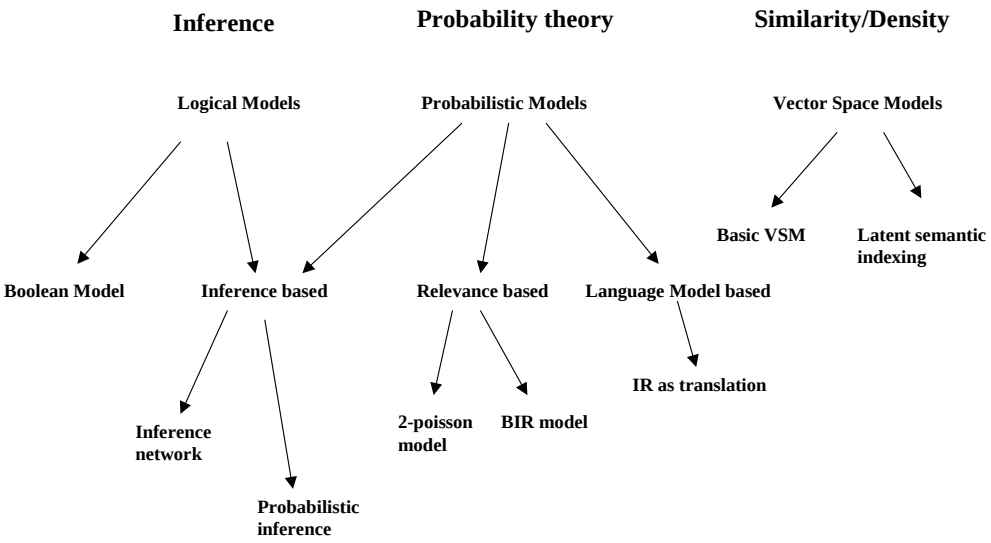

Figure 2.3. Taxonomy of IR models

model which is not based on relevance ranking. The rest of the models estimate the relative relevance of documents (ii) axiomatic vs. empirical models: One could say that both vector space and probabilistic models are statistical models, because they use word occurrence statistics. However, a vector space model is usually seen as a separate class because it does not employ probability theory to obtain the final document score. It is often argued that the vector space model is not a model in the strict sense (i.e. Crestani et al. (1998b)). VSM does not rely on an axiomatic model, of which the properties are well understood. That does not mean however, that the approaches that are based on sound probabilistic or logical models provide a more adequate description of the objects that we want to model and, eventually, the retrieval process that we want to optimize (Cooper, 1994) . That is mainly because, if we want to use these models for a real-life application, we usually have to make quite crude assumptions which lead to a simple model. If we do not make these simplifications, the model contains too many parameters, which cannot be estimated in a reliable way. Cooper argues that the PRP as such does not necessarily lead to a probabilistic model. Any IR technique which imposes an ordering on the documents based on some notion of relevance adheres to the PRP. But probabilistic models at least have clear assumptions (which might not be entirely true), ensuring that every step in the probabilistic inference process has a theoretical justification. We will compare the three basic classes by looking at the way in which the notion of relevance is operationalized in the corresponding framework.

The best known example of a retrieval model based on logic is the Boolean retrieval model. The query q can be expressed using index terms and operators from the Boolean algebra: conjunction, disjunction and negation. These logical operators have an intuitive set-theoretic semantics: each index term refers to a sets of documents indexed by that term. The AND operator restricts the query result to the intersection of two sets, the

OR operator yields the union and the NOT operator the difference between the sets. Van Rijsbergen has presented the Boolean model slightly differently: a document is relevant if the query can be derived from its set of index terms (and the closed world assumption) using the inference rules of propositional calculus: a document represented by the propositional formula d will be retrieved when its truth implies q. However, this retrieval model completely ignores the phenomenon of uncertainty which is inherent to IR (cf. section 1.3). It is the only retrieval model which is not based on the notion of relevance ranking. A Boolean retrieval function will divide the document set into two classes, one class contains the documents which support the boolean query i.e. for which $d \rightarrow q$ holds; the other class contains the documents that do not support the boolean formula. Both classes are opaque without any internal ordering. But the Boolean model is just one instance of the class of logical models. A new impulse was given to the development of these models by the publication of van Rijsbergen (1986). Van Rijsbergen shows that different retrieval models (Boolean, probabilistic) can be re-expressed as examples of computation of logical implication. He develops a non-classical (conditional) logic of which the semantics are expressed in probability theory. Section 2.4.5 will give a brief overview of IR models based on non-classical logics.

For the vector space model the relevance of a document d for a query q is defined as a *distance* measure in a high-dimensional space, therefore vector space models could also be called algebraic models. The distance measure actually serves as a metric to compute the *similarity* between queries and documents. In order to compute this similarity measure it is necessary to firstly project documents and queries in the high-dimensional space defined by the vocabulary of index terms.

The classical probabilistic models exploit the different distributions of terms in the class of relevant and the class of non-relevant documents. They calculate query term weights which are directly related to the *odds*[13] that the term in question is present in a relevant document. Recently, another probabilistic approach to IR which is based on statistical language models has proven quite successful. The intuition here is that the probability that a document is relevant with respect to a query can be modeled by the probability that the query is generated by a combination of two language models: a model estimated on the document in question smoothed by a model which is estimated on the complete document collection.

We do not claim to be exhaustive in our discussion of IR models. Apart from the three main model classes, IR systems can also be based on, for example, neural networks or genetic algorithms. In our opinion, these machine learning based approaches are more suitable for the information filtering or routing task, which include training data.

In the following sections each of the approaches to IR will be discussed in more detail. The interested reader can find more background information on models in Rijsbergen (1979), Salton & McGill (1983), Grossman & Frieder (1998), Sparck Jones & Willett (1997b), Crestani et al. (1998b), Frakes & Baeza-Yates (1992) or Baeza-Yates & Ribeiro-Neto (1999).

[13]The "odds" is a statistic which is frequently used in probabilistic models. The odds can be defined as $O(y) = P(y)/P(\bar{y}) = P(y)/(1 - P(y))$.

2.3.3. Common notation. In our discussion of the different models we will work with a common notation for variables as much as possible for readibility purposes. This has the consequence that in a few cases, the notation will differ slightly from the notation used in the original works. Documents and queries are often represented by an ordered list or vector of (weighted) terms. A document and a query will be represented by the arrow vector symbols $\vec{d}$ and $\vec{q}$ in the discussion about vector space models or by the random variables D and Q in the context of the probabilistic framework. Table 2.1 lists some other frequently used variable names. All models except the Boolean model try to predict

N	number of documents in the collection
V_c	number of unique terms in the collection (collection vocabulary)
V_q	number of unique terms in the query
V_d	number of unique terms in a document
tf	term frequency

Table 2.1. Variable names

the (relative) relevance of a document by applying a ranking function, which produces a partial ordering of the documents. We will use the theory neutral term *retrieval status value* (RSV) to denote the score, probability or other relevance estimate which is assigned by the ranking function to each (document,query) pair.

2.4. Logical models

Adopting the framework of logic has been an attractive avenue for the development of IR models. The well-defined theoretical properties of classical logical models are appealing but have their limitations, because they fail to model uncertainty, a central property of the IR problem. Recent work has shown that non-classical logics might very well bridge the gap.

2.4.1. Boolean model. The earliest IR systems were Boolean systems. Even today, a lot of commercial IR systems are based on the Boolean model. The popularity among users is largely based on the clear set-theoretic semantics of the model. In a Boolean system, documents are represented by a set of index terms. An index term is seen as a propositional constant. If the index term occurs in the document, it is true for the document, and following the closed world assumption, it is false if the index term does not occur in the document. Queries consist of logical combinations of index terms using AND, OR or NOT and braces. Thus a query is a propositional formula. Every propositional formula can be rewritten as a disjunctive normal form which can be efficiently evaluated for each document. The ranking function is thus a binary decision rule: if the formula holds for a document, it is considered relevant and retrieved. The Boolean retrieval model is very powerful, since in theory a query could be constructed which only retrieves the relevant documents, provided that each document is indexed by a unique set of index terms. However, without knowledge of the document collection it is impossible for a user to create such a query.

The conceptual clarity of Boolean systems is important for users. They know exactly how a query is evaluated, because the resulting documents will satisfy the Boolean constraint of the query. This gives the user a feeling of tight control of the retrieval function. However, Boolean systems also have considerable disadvantages:

(1) Since documents are modeled as either relevant or non-relevant, retrieved documents are not ordered with respect to relevance and documents that contain most query terms are not retrieved.
(2) It is difficult for users to compose good queries. As a result, the retrieved set is often too large or completely empty.
(3) The model does not support query term weighting or relevance feedback.
(4) Boolean systems display inferior retrieval effectiveness on standard IR test collections. (Salton & Buckley, 1988; Sparck Jones, 1999; Baeza-Yates & Ribeiro-Neto, 1999).

In the next two subsections we discuss two IR methods which are not based on logical models in the strict sense, but can be considered as Boolean systems with extra features. They do not take the frequency of index terms into account.

2.4.2. Co-ordination Level Matching. One way to remedy some of the disadvantages of the strict Boolean interpretation of queries is to model likelihood of relevance as the number of index terms that a query and a document have in common. The method of co-ordination level matching (CLM) presupposes that both query and documents are respresented as a set of index terms, so the query has no internal (Boolean) structure. For CLM, the retrieval status value is defined as the number of unique query terms found in the document[14]. The higher this number, the higher the co-ordination level. This approach has the advantage that the result set is ordered and that partially matching documents are retrieved, which might be desirable properties for naive users. The commercial IR system Muscat, for example, which claims to have a solid foundation in probabilistic models, uses coordination level matching as its primary ordering criterion. However, evaluation experiments (e.g., (Salton & Buckley, 1988)) show that the retrieval quality of CLM is inferior to the models which do explicit term weighting.

2.4.3. Proximity matching. Most commercial systems based on Boolean retrieval offer additional facilities to enhance the precision of a retrieval result. The use of standard Boolean queries on large document collections like the WWW is cumbersome, because usually a short query leads to a huge result set. Subsequent query refinement by additional terms combined with AND operators quite often results in an empty result set. Co-ordination level matching helps to some degree, but a more powerful method is the use of position information of index terms. Usually IR systems make use of an inverted index file of a document collection (cf. section 2.1). In principle we can also record the position of each index term in a document. This means that each occurrence of an index term in a document will be recorded as a *posting*[15] in the index, which will increase the

[14] In fact CLM can be seen as a simple form of vector-based retrieval, cf. section 2.5.

[15] An index is usually stored in posting files. Posting files contain the references of an index term in sequential order. These references are contained in so-called postings. A posting usually contains a document reference and a term weight, but sometimes contains position or type information as well.

size of the index several times (this is in fact the type-token ratio). If full position information of index terms is available, an IR system can compute the distance between index terms and thus support *exact phrase* queries or *proximity queries*. The former searches for exact occurrences of a phrase, the latter relaxes the constraint of strict adjacency and will retrieve documents where the index terms occur within a 'window'. One step further is to calculate a relevance estimate based on the distance between query terms in the document. This method is called *cover density ranking* (Clarke et al., 1997) and has been implemented in the IR system of the university of Waterloo. The cover density ranking is a secondary ordering criterion, which is applied *after* ranking by co-ordination level. The method is especially suitable for short queries. For each document the shortest document fragments[16] that satisfy the (Boolean) query are determined. These fragments are ranked inversely proportional to their length in descending order. Subsequently the document score is computed by summing the fragment scores. The contribution of the n^{th} fragment is down-weighted by a factor γ^n where γ is typically a value between 0.5 and 1. Clarke at al. claim that this method yields retrieval effectiveness results which are competitive to systems which exploit global term statistics like inverse document frequency, while satisfying the co-ordination level ranking which is appreciated by most users, because it is a very simple intuition. The Waterloo system (Clarke et al., 1997) has at least proven to be a very effective tool for the manual *ad hoc* runs[17] e.g., at TREC-7.

2.4.4. Alternative set theoretic models. There heve been several proposals to base IR systems on alternative set-theoretic models in order to cope with the uncertainty problem (cf. section 1.3). We will briefly present two of these alternatives for the Boolean model, which define new semantics for set-theoretic operators.

2.4.4.1. *Fuzzy set model.* Several models for IR based on fuzzy set theory (Zadeh, 1965) have been proposed (cf. Salton (1989) for an overview). In a fuzzy set, elements have a gradual membership value. Unlike in Boolean models, where term-document membership values are binary, fuzzy membership values range between 0 and 1. The advantage of this approach is that degrees of belief can be encoded. For example, one could compute a term-term correlation matrix and add terms that are correlated to the terms of a particular document to the representation of that document with the correlation as membership value. Evaluation of a query in fuzzy logic differs in the semantics of the intersection(and) and union(or) operators which are expressed as respectively the minimum or maximum membership value. Fuzzy IR models have not been tested on large test collections.

2.4.4.2. *Extended Boolean model.* Most queries issued at current WWW search engines are fairly short: 2-3 terms. Search engines are often based on Boolean retrieval, implicitly assuming AND operators between query terms. However, in many cases, the actual information need of a user cannot really be captured easily in Boolean logic. Not every term is equally important, which cannot be expressed in a Boolean query. Simple *tf.idf*

[16] A fragment is determined by a begin and end position in the document, the original paper uses the term "substring".

[17] The ad hoc task is a standard IR evaluation task at the Text REtrieval Conference. Ad hoc refers to a single query without any prior relevance information in contrast with the routing task, which models a long standing query (or profile) for which relevance information can be gathered.

term weighting schemes[18] could remedy this, but these schemes often have a too weak co-ordination since they lack Boolean operators. The extended Boolean model (Salton et al., 1983) integrates term-weighting and distance measures into the Boolean model. Firstly, like in Fuzzy retrieval, index terms can be weighted between 0 and 1 (for example by using a normalized *tf.idf* scheme). Secondly, the Boolean connectives have a new semantics, they are modeled as similarity measures based on non-Eucledian distances in a V_c-dimensional space. The extended Boolean model has been further generalized in the p-norm model. Here the semantics of the OR and AND connective contains a parameter p. By varying the parameter p between 1 and infinity, the p-norm ranking function varies between a vector space model like ranking and a (fuzzy) Boolean ranking function. In principle p can be set for every connective.

Despite their conceptual appeal, extended Boolean models have not become popular. One of the reasons could be that the models are less perspicuous for the user. Queries still have the form of a Boolean formula, but with changed semantics. Many users prefer not to spend a lot of time to compose a structured query. For long queries, a vector space or probabilistic system is to be preferred. For two-word queries a Boolean AND query is usually but not always sufficient. Extended Boolean systems in combination with sophisticated user interfaces which give feedback on term statistics might be attractive especially for a more robust handling of short queries.

2.4.5. Models based on non-classical logic. We have seen that classical logic fails to model the uncertainty which is inherent to IR. However, the logical approach to IR got a new impulse by van Rijsbergen, who showed that non-classical logics can form the framework for IR (van Rijsbergen, 1986). He demonstrated that several classical IR models can be re-expressed as computation of logical implication. The basic notion is that logically spoken, retrieval can be expressed by the implication $q \leftarrow d$. However, because of the uncertainty involved, this is not a material implication in first order logic, but requires a non-classical, conditional logic.

Van Rijsbergen's work stimulated renewed interest in logical and uncertainty models for IR. Crestani et al. (1998a) gives a recent overview of uncertainty models and logics for IR. The logical approach to IR allows integration of external knowledge sources like user's beliefs (Nie & Lepage, 1998) and can be used to model document structure and relationships between information objects (Rölleke & Fuhr, 1998) or multimedia documents (Meghini et al., 1998). Partial implementations of some of these models exist. There have, however, not been large scale evaluations of these models. One of the problems is their computational complexity. There is one exception, Crestani et al. (1996) describes an evaluation of the Logical Imaging model at TREC-4. This model seems to yield improvement over classical *tf.idf* approaches on small collections (Crestani, 1998). However, the model performed disappointingly on the TREC collection. A failure analysis showed that the major reason was a lack of effective document length normalization. A second reason was that the model had to be approximated in order to scale up to the TREC collection and that these approximations made the model apparently less effective (Crestani, 2000).

[18] *tf.idf* refers to the basic vector model where term weights are proportional to the term frequency and inverse document frequency, cf. section 2.5

The framework of probabilistic inference has been extended by Wong and Yao, who show that all classical IR models (including vector space models) can be re-expressed as probabilistic inference, thus showing the relationships between these models (Wong & Yao, 1995).

2.5. Vector space models

The first ideas for a text representation based on weighted term vectors were already put forward in the late 1950s by Luhn (1957) . He proposed to represent both information enquiries and documents as a set of concepts (*"notions"*) stored on punched or magnetic tape and envisaged a statistical matching process:

> "Since an identical match is highly improbable, this process would be carried out on a statistical basis by asking for a given degree of similarity".

It is exactly the notion of similarity which is characteristic for the vector space model (VSM) approach. In contrast with the Boolean model where the matching function is based on an *exact match*, the Vector Space approach starts from a more fine grained view on relevance estimation. A VSM based system determines the similarity between a query representation and a document by interpreting both (vector) representations as points in a V_c-dimensional space and taking a vector distance measure as similarity metric and thus as relevance predictor. The similarity is assumed to be correlated with the probability of relevance of the document. So, when we order documents according to their similarity to the query, the system adheres to the Probability Ranking Principle (cf. section 2.1.3).

The ideas of Luhn were further developed by Gerard Salton, first at Harvard, later at Cornell University. Salton developed VSM into a powerful retrieval framework, embodied in the SMART project (Salton's Magical Automatic Retriever of Text) that ran from 1961 until 1996 (Lesk et al., 1997). The work of Salton has been very influential. For years he was the preeminent figure in the IR community. He has authored several textbooks (e.g., Salton & McGill, 1983; Salton, 1989), and numerous papers. Many IR researchers have worked and still work in the vector space paradigm, partly because of the free availability for researchers of the SMART system.

We will discuss the basic vector space model and its assumptions in sections 2.5.1-2.5.2. Some of the experiments in chapter 6 ar based on a vector space model. Our goal is therefore to explain the components of frequently used vector space models like $ntc.atn$ and $Lnu.ltu$ in some detail. The motivation behind these models is scattered around many different papers and reports. Subsequently we will discuss several more advanced models, which are based on a vector space representation: latent semantic indexing (section 2.5.3), the generalized vector space model (GVSM) (section 2.5.4) and the construction of similarity thesauri (section 2.5.5). The latter has an important appplication for cross-language information retrieval.

2.5.1. Basic formalization. In the vector space model, documents and query are represented by a vector:

$$\vec{d_k} = (w_{k,0}; w_{k,1}; \ldots; w_{k,V_c}) \tag{6}$$

$$\vec{q} = (w_{q,0}; w_{q,1}; \ldots; w_{q,V_c}) \tag{7}$$

Here, $\vec{d_k}$ represents document d_k and $\vec{q}$ represents the query, $w_{k,i}$ is the term weight for the i^th term of the indexing vocabulary for document k. Note that we have defined a mapping of document and queries into a V_c-dimensional space. Typically V_c ranges from the order of 10^4 for a small text collection to 10^6 for large text collections. Consequently both document and query vectors will be very sparse, this means that most of the term weights for document and query vectors are equal to zero.

All ranking functions of the vector space family are based on the inner product: (RSV = Retrieval Status Value):

$$\text{RSV}(\vec{q}, \vec{d_k}) = \sum_{i=1}^{V_c} w_{q,i}.w_{k,i} \tag{8}$$

Suppose we take a simple presence/absence term-weighting scheme (term weight is either 0 or 1), then equation (8) describes coordination level matching (cf. section 2.4.2), i.e. CLM rewritten as an inner product.

However, term weights are of course not restricted to a binary scheme, in principle they can take any (positive) value. The classical example of a term weighting scheme developed by Salton is *tf.idf*: a term weight which is proportional to the frequency of occurrence within the document and inversely proportional to the number of documents the term occurs in. We will discuss these term weighting variants later. For now, we want to concentrate on the vector representation of documents and queries and simply presuppose a certain term weighting scheme with a *tf* and *idf* component.

Vector length normalization (cosine similarity). An important problem is formed by heterogeneous documents lengths. Consider, for example, two documents. One of them is one page long and contains 90% of the query terms. Another document is 10 pages long and also contains 90% of the query terms (with 10 times higher frequencies) but apart from these many additional terms. In this case one could argue that the shorter document is more similar to the query than the long document (since the shorter document is more focused on the query concepts), but the longer document will have the highest RSV as defined in (8). One elegant way to normalize scores is to apply *vector length normalization*. When we apply vector length normalization, we can also give a more intuitive explanation of the inner product as the basis for the matching function; the length normalized inner vector product corresponds to the cosine of the angle between the vectors:

$$\text{RSV}(\vec{q}, \vec{d_k}) = \cos(\vec{q}, \vec{d_k}) = \frac{\vec{q} \times \vec{d_k}}{||\vec{q}|| \, . \, \left|\left|\vec{d_k}\right|\right|} = \sum_{i=1}^{V_c} \frac{w_{q,i}.w_{k,i}}{\sqrt{\sum_{i=1}^{V_c} w_{q,i}^2} \cdot \sqrt{\sum_{i=1}^{V_c} w_{k,i}^2}} \tag{9}$$

Equation (9) defines the so-called *cosine normalization function*. The inner product computation can take advantage of the fact that only the products for the index terms where the query term and the document term weight do not equal zero add to the RSV. Given this observation we can rewrite (9) to

$$\text{RSV}(\vec{q}, \vec{d_k}) = \cos(\vec{q}, \vec{d_k}) = \sum_{i=1}^{V_q} \frac{w_{q,i}.w_{k,i}}{\sqrt{\sum_{i=1}^{V_q} w_{q,i}^2} \cdot \sqrt{\sum_{i=1}^{V_{d_k}} w_{k,i}^2}} \tag{10}$$

where V_q refers to the number of unique query terms, and T_{d_k} to the number of terms in document i. The main summation is only based on the query terms and not on the full indexing vocabulary. Usually the document length normalization factor is computed off-line and included in the term-weight posting.

Of course, vector distance measures in a V_c-dimensional space are not limited to the cosine. Other geometric similarity measures like the Eucledian distance: $\left|\left|\vec{q} - \vec{d}\right|\right|$ could be used, but the cosine has the advantage of being insensitive for substantial differences in document length[19]. The inner product also enables an efficient ranking algorithm. The geometric interpretation of the vector space model is presented and extended in Salton et al. (1975), where the author shows that relevant documents tend to cluster together in the multidimensional space.

Salton & Buckley (1988) discern two main issues in the design of automatic text retrieval systems:

> "First, what appropriate content units are to be included in the document and query representations? Second, is the determination of the term weights capable of distinguishing the important terms from those less crucial for content identification."

Salton & Buckley experimented with more complex document content representations, by including e.g., related terms, phrase or thesaurus terms. However, none of these methods yielded a significant improvement, possibly because many complex index terms were too infrequent (Salton & Buckley, 1988). They had more succes with their experiments with different term-weighting schemes based on term statistics, we will discuss the main components of these schemes.

The tf.idf formula. A first step to improve retrieval effectiveness was to include the frequency of occurrence of an index term in a document in the ranking formula, usually referred to as term frequency *tf*. This factor captures the intuition that whenever a term is mentioned more often in a document, it will probably be a central term for the document, and thus a good relevance predictor. Later, it was found that a factor inversely proportional with the number of documents in which the term occurs at least once is also a quite effective term weighting factor. This factor is usually referred to as the inverse document frequency or *idf*. The inverse document frequency is based on statistics of the term in the collection and is therefore also reffered to as (an instance of) collection frequency weighting (Robertson & Walker, 1994). The *idf* factor (which is usually incorporated as a logarithmic value) captures the intuition that the more general a term is, the

[19] For $||\vec{q}|| = \left|\left|\vec{d}\right|\right|$ the Eucledian distance is a monotonic function of the cosine and thus will induce the same document ranking.

poorer it discriminates. As an illustration: function words like determiners, auxiliaries and prepositions are very common, giving them a very low idf weight. This corresponds well with the fact that they do not carry content. Usually, function words are removed from the indexing vocabulary in order to compact the index and increase retrieval speed.

The combination of both term weighting components together with cosine normalization is the prototype for what is frequently referred to as the *tf.idf* vector space model. Substituting w_i

$$w_i = tf_i \cdot idf_i \tag{11}$$

for the query and document weight in equation (10) yields:

$$\text{RSV}(\vec{q}, \vec{d_k}) = \cos(\vec{q}, \vec{d_k}) = \sum_{i=1}^{V_q} \frac{tf_{q,i} idf_k . tf_{k,i} idf_i}{\sqrt{\sum_{i=1}^{T_q} (tf_{q,i} idf_i)^2} \cdot \sqrt{\sum_{i=1}^{T_{d_k}} (tf_{k,i} idf_i)^2}} \tag{12}$$

Although a strict application of the vector space metaphor leads to an equivalent term weighting formula for both query and documents like in equation (12), specific term weighting functions for query and documents yield better results.

Improved term weighting strategies. Salton and Buckley have experimented with many different variants of the basic vector space model. The development of these new variants was driven by analysis of results on test collections rather than motivated by theoretical insights. The different term-weighting schemes for vector space models are usually referred to by a six letter code. The letters refer to the term frequency weight, the collection frequency weight and the normalization method for the document and the query. For example term weighting formula (12) is represented by the code $ntc.ntc$. Unfortunately, these codes have not been used consistently in publications. The code scheme used by the SMART system is not consistent with the scheme described in one of the principal publications about term weighting schemes in VSM (Salton & Buckley, 1988). Appendix A presents these letter codes in more detail and explains the problem that was introduced by the inconsistent use of codes. In the rest of this thesis we will use the SMART system codes for VSM term weighting formulas.

The space of possible term weighting combinations which is defined by all possible combinations of term weighting components has been explored[20] by performing extensive evaluations on different test collections: CACM, CISI, Cranfield, INSPEC, MED and NPL (Salton & Buckley, 1988). The collections are very small according to current standards but exhibit an interesting variety in average query and document lengths. The article reports results on eight 'well-known' term-weighting methods: including *tf.idf*, probabilistic and coordination level methods. Two term weighting configurations *ntc.atn* and *nnc.atn*[21] performed consistently well on all but one of the test collections. The exception is the NPL collection with relatively short documents and queries. Here the probabilistic method *ann.bpn* performs best. As an example we give the full document

[20] This approach of defining a space of possible term weighting factor combinations and doing an exhaustive evaluation of all possible combinations has also been carried out by Zobel & Moffat (1998).

[21] We converted the notation published in (Salton & Buckley, 1988) to the more commonly used sSMART notation.

ranking formula for *ntc.atn*:

$$\text{RSV}(\vec{q},\vec{d_k}) = \sum_{i=1}^{V_q} \frac{(0.5 + 0.5\frac{tf_{q,i}}{\max(tf_q)})\log\frac{N}{df_i} \cdot tf_{k,i}\log\frac{N}{df_i}}{\sqrt{\sum_{i=1}^{T_{d_k}}(tf_{k,i}\log\frac{N}{df_i})^2}} \tag{13}$$

Salton and Buckley conclude the following:

Query Vectors: For short vectors with term frequency variance, the augmented normalized term frequency weight (*n*) is preferred. For longer queries, the raw term frequency performs better. For the collection frequency weights take either the *f* or *p* factor. Normalization will not have any effect on ranking

Document vectors: For document vectors the same arguments hold, except that length normalization can really improve results in case of a collection with large variance in document length

At the time of these experiments, researchers used different test collections and consequently it was hard to compare results. In some cases superiority was claimed on the basis of just one small test collection. It was this cumbersome situation which motivated the National Institute of Standards and Technology (NIST) to start the TREC initiative, which has a far more rigorous evaluation with large test collections and blind tests. We will discuss evaluation matters in more detail in chapter 4.

The influence of TREC on the development of term-weighting schemes. The SMART system was improved further during the annual TREC conferences started in 1991 (cf. section 4.2). An interesting overview of the impact of TREC on the performance of the SMART retrieval system is given in Buckley et al. (1999). The mean average precision (cf. section 4.3.4) on the TREC-1 task has improved with 55% in the course of six years (cf. table 2.2). The basic SMART system used at TREC-1 used *ntc.ntc* weighting, the standard *tf.idf* (cosine) formula. From TREC-1 on, each year saw a major improvement until around TREC-4. Often techniques were incorporated that had proven useful for other participants in the previous TREC. The major improvements were: replacement of the augmented normalized term frequency by the log of the raw term frequency, automatic query expansion and pivoted document length normalization. These techniques will be explained below.

TREC nr	method	Mean average precision
TREC 1	ntc.ntc	.2442
TREC 2	lnc.ltc	.3056
TREC 3	lnc.ltc-Exp	.3400
TREC 4	Lnu.ltu-Exp	.3528

Table 2.2. Improvement of SMART on TREC-1 task over the first TREC years

Logarithmic term frequency. In TREC-2 the term frequency component was 'dampened' by a logarithmic function because the influence of term frequency was considered too large (empirical motivation).

Automatic query expansion. Though automatic relevance feedback (cf. section 3.1) already had been studied in pre-TREC experiments, by simply assuming that the top N documents of a retrieval run are relevant, it was not generally applied, because it did not yield a consistent improvement of retrieval performance on the test collections in use before TREC. Buckley et al. (1995) describe that the positive experiences with automatic query expansion on the TREC2 routing task and the good results on the TREC2 ad hoc task by the UCLA and CMU groups triggered a renewed interest in automatic query expansion. The Cornell approach is based on massive expansion with about 300 terms selected from the top N documents. They claim that the success of automatic expansion on the TREC collection is due to two factors: (i) better weighting algorithms (ii) a large collection and therefore more relevant documents per query. Both factors increase the probability of relevance of the documents in the top N of the hit set. This probability of relevance is crucial, because the automatic query expansion approach assumes that the feedback documents are relevant. See also section 3.1 for a more elaborate discussion of automatic query expansion techniques.

Pivoted document length normalization: the $Lnu.ltu$ formula. Already at the time of TREC1, the SMART group observed that applying cosine normalization on both queries and documents was not optimal for TREC collections. "*idf* in both [queries and documents] ends up over-weighting medium terms" Buckley (1999). It was also realized that the cosine normalization on the documents poses practical problems when collections are dynamic. Adding documents to the collection would in principle call for a recalculation of all document vectors, because the document frequencies are updated. Another problem was formed by the long documents in the TREC collections (in comparison to the small abstract based collections used thus far). These documents (with a considerable amount of misspellings) contained so many unique terms, that they were unretrievable because of the low term weights, induced by the cosine normalization. At TREC-4 a new document normalization scheme was introduced, which addressed these problems. Again the SMART group observed that the SMART term weighting scheme was less effective than the methods applied by the Okapi (cf. section 2.6.1.2) and INQUERY group (cf.section 2.6.2.1). A detailed analysis by Singhal showed that the SMART system retrieved less *long* relevant documents than the other systems (c.f. Buckley et al., 1996; Singhal et al., 1996). Singhal compared the document normalization techniques which were used by SMART, Okapi and INQUERY. The analysis showed that the assumption that the a-priori probability of relevance does not depend on document length is not valid for the TREC collection. However, normalizing the RSV with respect to document length is the prime objective of the cosine document length normalization in the SMART weighting scheme. The actual probability of relevance of a document as measured on the TREC test collection shows a more or less linear function of the document length, both probability curves can be approximated by straight lines and cross each other. Singhal proposed to apply a transformation on the document length normalization function in order to boost the scores for longer documents and decrease scores for shorter documents, by "tilting" around the pivot pount where both curces cross. The transformation function (pivoted normalization) contains two parameters: the *slope* and the *pivot*. Both parameters have to be "trained" (or tuned) on a previous collection. The resulting term

weighting scheme is referred to as $Lnu.ltu$ (where the L and l represent two term frequency factor variants based on normalization on average term frequency) and the u stands for pivoted *unique* normalization), and has proven to be very successful. Its performance is comparable to, for example, the BM25 weighting scheme used by the Okapi group (cf. section 2.6.1.2). Table A.1 in appendix A lists the new term weighting components with their corresponding letter encoding. Note however, that the original idea of cosine normalization has completely been abandoned in the current state-of-the-art vector space systems. The $Lnu.ltu$ system is effective but we consider its motivation not so elegant. Equation (14) shows the full ranking formula for $Lnu.ltu$:

$$\text{RSV}(q, d_k) = \sum_{i=1}^{V_q} \frac{(1 + \log tf_{q,i}) \cdot \log \frac{N}{df_i}}{(1.0 - s)p + s \cdot V_q} \times \frac{\left(\frac{1+\log(tf_{k,i})}{1+\log(\sum_{i=1}^{V_q} tf_{i/L})}\right) \cdot 1.0}{(1.0 - s)p + s \cdot V_d} \tag{14}$$

where V_q and V_d are the number of unique terms in the query and the document respectively, p and s are pivot and slope and L is the indexing vocabulary size.

2.5.2. Term dependence. Like most other IR models which have actually been implemented and tested on large collections, VSM presupposes *term independence*. In an Eucledian space, this means that it is assumed that all terms are pairwise orthogonal. Since concepts in a document (and thus index terms) are often semantically related to each other (and thus occur more often together than would be expected by chance), term independence is an unrealistic assumption. Common practice in VSM based approaches is to circumvent the dependency problem, by simply assuming that the axes of the vector space are orthogonal. The resulting model is easy to implement, extend and conceptualize.

One could build a vector based IR model which takes term dependence into account, by assuming independence between pairs of triplets of terms (Salton, 1989; Raghavan & Wong, 1986; Yu et al., 1983). Although the independence assumptions are less strong for these models, they also require many more parameters to be estimated. And ususally, there is not enough relevance data avalailable to estimate these parameters with a reasonable accuracy. As Fuhr (1992) observes:

> "As a consequence, experimental evaluations showed that the gain from improved independence assumptions does not outweigh the loss from increased estimation errors."

However, there are other ways to exploit dependency between terms for the improvement of IR systems. As we have seen in section 2.1.1, one of the most important problems in IR is the terminology mismatch between queries and relevant documents. Suppose now that an information structure which models the semantic links between terms is available, then this structure could be used to remedy the terminology mismatch and thus improve recall. One way to build such an information structure is to assume that term co-occurrence in documents (term dependence) corresponds to semantic similarity. There are two approaches that try to leverage the co-occurrence information in order to improve recall. The first approache tries to to normalize terminology by mapping related content terms onto one single concept: *concept indexing*. The second approach uses the information structure (or the underlying term co-occurrence information) for query expansion. We will present some of the former approaches in the following subsections,

some examples of the latter approach will be presented in section 3.1. But in fact both techniques are related.

2.5.3. Latent Semantic Indexing. The vector space model has inspired several researchers to apply techniques from linear algebra. One such a technique is Singular Value Decomposition. SVD is very close to the Principle Components Analysis dimension reduction technique used in statistical multivariate analysis, but contrary to PCA which can only be applied on rectangular matrices, SVD can be applied to any matrix. The intuition here is that when a high dimensional data set contains a fair bit of dependency, we could *approximate* the same data set with a model with fewer dimensions. Co-occurrence of terms is an indication of dependence. If terms cooccur frequently in a document, they might be semantically related. SVD will project cooccurring words onto the same dimension, and independent terms onto different dimensions. The application of singular value decomposition to document-by-term matrices is thus called *indexing by latent semantic analysis (LSA)*, or simply *latent semantic indexing* (LSI). LSI has the effect of clustering (actually projecting) synonyms in the same dimension, which has the very desirable effect that the recall of a query will be improved.

We will give a brief introduction on the mathematical backgrounds of the technique and discuss the application of LSI in IR. A more detailed account can be found in Deerwester et al. (1990) and Manning & Schütze (1999). Suppose we have a document-by-term matrix $A_{l\times d}$, where l is the number of unique index terms in the collection and d is the number of documents. For each matrix of that type, there exists a unique decomposition into the product of three matrices T, S and D^T (the transposed matrix D):

$$A_{l\times d} = T_{l\times n} S_{n\times n} (D_{d\times n})^T \tag{15}$$

where n is $\min(l, d)$. The decomposition has the following properties

(1) columns of T and D are orthonormal, i.e. they have unit length and are orthogonal to each other
(2) The matrix S is a diagonal matrix with non negative singular values in descending order

SVD can be explained as a rotation of a V_c-dimensional space which projects the data into a new space where the highest variation among the data points (i.e. the documents) is along the first dimension, the second highest variation along the second dimension and so on. However, the most important property of this decomposition is that when we restrict the matrices to a lower dimension, i.e. by reducing n to k where $k << n$ (which in practice means deleting columns or rows from the respective matrices) the resulting matrix A' is the optimal approximation of A in k dimensions. In most cases, k is in the order of 100-400, which means a reduction of the order of 1000. It is quite surprising that this vector space model with such a low dimension still works effectively and even leads to a more effective IR system.

The LSI approach has been evaluated by a number of groups, notably at Bellcore and CMU. The Bellcore group has reported results on different collections. But these results are not consistent. Deerwester et al. (1990) reports a 13 % improvement in precision on the small MED collection (1033 documents, 30 queries), and a decrease in average precision (all figures compared with a SMART VSM baseline system) on the CISI collection

(1460 documents and 35 queries). The largest collection on which an evaluation has been published is TREC-3 (Dumais, 1995). This evaluation showed a 5% improvement in average precision of LSI versus plain SMART (0.2393 vs 0.2220). The authors argue that the disappointing result is due to the fact that TREC topics are long. For long topics, recall enhancing techniques like LSI would not be able to produce a marked improvement. We think this is only partly true, first of all because average precision is a measure that contains components of both precision and recall. Secondly because other groups have used co-occurrence in their IR system and yielded much better results. For example, the SMART group reports an average precision of 0.2842 with a plain vector space scheme (lnc.ltc) and 0.3419 for a standard pseudo-relevance feedback approach (cf. section 3.1). As Dumais argues, part of the absolute differences are due to differences in term-weighting, but the relative improvement of the pseudo feedback method is much more substantial than the LSI approach.

Despite the elegant idea, LSI did not become part of mainstream IR. We can think of several causes:

- LSI is computationally very demanding, the complexity is quadratic in the rank of the document by term matrix. Every query requires a high dimensional matrix computation as a pre-processing step as well. The same holds for each new document that one wants to add to the index. As such it does not scale up well.
- The resulting dimensions are hard to characterize,
- Term co-coccurrence can be exploited in a *cheaper* way by applying pseudo-relevance feedback
- LSI yielded poor results at the TREC evaluation conference.
- A more technical objection to SVD is the fact that it is designed for normally-distributed data. Recently, a probabilistic version of latent semantic indexing has been proposed (Hofmann, 1999). Unfortunately, this approach has only been tested on small test collections (though with good results).

2.5.4. Generalized Vector Space Model. Another attempt to remedy dependency is the *Generalized Vector Space Model (GVSM)*, (Wong et al., 1986, 1987). Instead of simple terms GVSM takes "generalized terms" or so called *minterms* as basic indexing units. A minterm in L terms is defined as a set of binary term weights for each term in the collection. A collection with vocabulary size of L terms yields 2^L possible minterms corresponding to all possible patterns of co-occurrence. Now both queries and documents are mapped into this space. This means that documents with the same terms are mapped to the same minterm. The minterm vectors are linearly independent (and thus form a basis) and orthogonal, while the index terms themselves are allowed to to dependent. Wong et al. tested GVSM on some small collections, producing slightly better results than standard VSM based on a binary indexing scheme. We think that, given the computational complexity of the model, these small improvements are not convincing.

GVSM has been simplified to use just documents as the basis for the vector space. This idea is sometimes called the *dual space*. Starting point for the standard VSM is the term document matrix $A_{L\times N}$ where the L rows refer to the indexing vocabulary and the N columns represent document vectors. VSM is based on taking the N columns as axes in the V_c-dimensional space. Simplified GVSM takes the L rows as axes in the V_c

dimensional space. These vectors can subsequently be used for measuring term-term similarity. Usually one uses a moderate sized document collection to produce matrix A. Subsequently queries and test documents are mapped into this space, by applying the transformation $\vec{d'} = A^T\vec{d}$. For a retrieval run, the cosine similarity $\cos(\vec{d'}, \vec{d'})$ has to be computed for every document. The latter operation has complexity $O(n)$ per document where n is the number of documents in the training collection. GVSM might be computationally less demanding than LSI, it is still quite resource consuming, making it an unfeasible option in interactive environments. Yang et al. (Yang et al., 1997, 1998) have carried out experiments with GVSM, in a comparison with LSI and standard VSM (SMART ltc weighting plus pseudo-relevance feedback on top 20 documents). The study focusses on Cross-Language Information Retrieval, but some monolingual results are reported on two test collections: the MEDLARS collection (1033 documents and 30 queries) and the UNICEF test set (1121 documents and 30 queries, for which complete relevance judgements were produced). The study shows that GVSM has the best performance. We think that the choice for small collections was partly motivated by the computational complexity of both GVSM and LSI. In large collections like TREC, standard VSM with pseudo-relevance feedback is still the method to beat, with a quite acceptable query response time.

2.5.5. Similarity thesaurus. A GVSM-related technique has been developed at ETH Zürich. Qiu (1995) describes a method to do query expansion based on a *similarity thesaurus* (Schäuble, 1989). A similarity thesaurus is similar to the GVSM vector space, where terms are indexed by documents. Thus the "meaning" of a term is represented by a weighted vector of documents. In the GVSM case, these weights are computed by first indexing the documents by terms using SMART and then transposing the matrix. Qui takes the more logical step to apply the term weighting schemes known from SMART in a more general fashion: as feature weighting schemes. He applies ntc weighting to produce *document weights*, where the *within document term frequency* and *inverse document term frequency* components of ntc are replaced by *within item (term) frequency* and *inverse item (term) frequency*. This different approach stems from a clear intuition:

> "It is also worth noting that the weight of a term representing a concept discussed in a document is not identical with the weight of the document representing (part of the) meaning of a term: the fact that a term describes a document well does not necessarily mean that the document represents properly the meanings of this term."(Qiu, 1995)

The resulting similarity thesaurus can subsequently be used for query expansion. The expanded queries can be evaluated by a conventional IR system. Expansion is of course especially useful for short queries. The results of experiments with queries consisting of only the description field on the TREC4 collection are summarized in table 2.3. Query

method	standard VSM	sim. thes. based exp.	local feedback	combination
map	0.1005	0.1523	0.1571	0.1691

Table 2.3. Comparison of query expansion methods (mean average precision)

expansion based on the use of the similarity thesaurus shows a marked improvement in

average precision, though slightly lower than the pseudo-relevance feedback approach described in (Buckley et al., 1995). Combination of both approaches performs even better, which shows that both methods capture different associations. The similarity thesaurus is based on global associations, whereas the local feedback approach (see also section 3.1) captures query specific (local) associations. In terms of efficiency, a similarity thesaurus can be computed off-line, and therefore can be applied immediately for expansion, it does not need a first pass retrieval run. Experiments on full TREC topics show a much smaller improvement (5%), indicating that the method is only useful for short queries. However, local feedback is still effective for full TREC topics. An interesting fact is that a similarity thesaurus built on 50% of the documents performs nearly as well as one built on the full collection. This gives some indication of the usefulness of a similarity thesaurus which has been trained on a (partly) different collection.

2.6. Probabilistic models

In section 1.3, we discussed the IR task under the title "Dealing with uncertainty". In short, this refers to the problem that it is difficult to distill the meaning from a search request or document and to infer whether a document is relevant for a request. In the previous section we have seen that term statistics can serve as an effective means to weight the importance of a term. However, the specific term weighting schemes of VSM have a rather heuristic basis. Probability theory has proved to be a more principled avenue to deal with uncertainty. The (classical) probabilistic takes the relevance relation as starting point, and uses term statistics for the estimation of parameters in the model. We will discuss three classes of probabilistic models in the following sections:

(1) *Probabilistic relevance models* try to estimate the relevance of a document directly based on the idea that query terms have different distributions in relevant and non-relevant documents.
(2) *Inference based models* apply Bayesian inference for the computation of a relevance score.
(3) *Generative probabilistic models*, also called *language models* as usually applied in automatic speech recognition systems, can also very fruitfully be applied for IR.

Most of our experiments in the chapters 5, 6 and 7 are based on language model based IR systems. Note that the relationship between probabilistic relevance models and generative probabilistic models and their respective properties are discussed in more detail in chapter 7.

2.6.1. Probabilistic relevance models. The first probabilistic model was already presented by Maron & Kuhns (1960). They proposed to base the ranking formula of an IR system on the application of probability theory. The easy part of developing probabilistic models is to apply probability calculus in order to reformulate the probability function into a simplified form, for example, by leaving out components which are not dependent on the document. But the resulting models contain a large number of parameters that have to be estimated, which is not always easy or even feasible in the case of more complex, refined models like the 2-Poisson model (see section 2.6.1.2). In the following subsections (2.6.1.1- 2.6.1.2) we discuss some well known probabilistic

models like the BIR model, the Robertson/Sparck Jones formula and the Okapi family of formulas which has its roots in the 2-Poisson distribution. An important aspect which distinguishes these models from, for example, the vector space family of models is that the models presuppose relevance information. In the case of *ad hoc* queries however, no relevance information is available. In this case the BIR model is equivalent to inverse collection frequency weighting. On the other hand, if relevance information is available (for instance in a *routing* task) the information that a certain document is relevant can immediately be used to improve the parameter estimates.

For a comprehensive overview and comparison with other probabilistic relevance models we refer to Crestani et al. (1998b), Fuhr (1992) and Rijsbergen (1979).

2.6.1.1. *Binary Independence Retrieval model.* An important family of probabilistic models is derived from the so-called *binary independence retrieval (BIR) model*. The basic idea is that term distributions are different for relevant and non-relevant documents. The basic BIR model only regards term presence or absence, so every document can be described with a binary term weight vector. The goal is now to derive a formula which estimates the probability that documents which can be described with a certain binary vector $\vec{d}_k$[22] are relevant for a certain binary query vector $\vec{q}$. Computing the odds of relevance of a document and apply Bayes' theorem yields:

$$O(\mathrm{R}|\vec{q},\vec{d}) = \frac{P(\mathrm{R}|\vec{q},\vec{d})}{P(\bar{\mathrm{R}}|\vec{q},\vec{d})} = \frac{P(\mathrm{R}|\vec{q})}{P(\bar{\mathrm{R}}|\vec{q})} \cdot \frac{P(\vec{d}|\mathrm{R},\vec{q})}{P(\vec{d}|\bar{\mathrm{R}},\vec{q})} \tag{16}$$

Now the linked dependence assumption can be applied, which says that the ratio of the probability that a document term vector $\vec{d}$ occurs in the relevant or non-relevant subset of documents can be computed by taking the product of the individual ratios of the individual terms of $\vec{d}$:

$$O(\mathrm{R}|\vec{q},\vec{d}) = O(\mathrm{R}|\vec{q}) \prod_{i=1}^{V_c} \frac{P(w_i|\mathrm{R},\vec{q})}{P(w_i|\bar{\mathrm{R}},\vec{q})} \tag{17}$$

Because these are binary vectors, the product can be split into two products, the first dealing with the terms that occur in the document, the second covering the absent terms:

$$O(\mathrm{R}|\vec{q},\vec{d}) = O(\mathrm{R}|\vec{q}) \prod_{w_i=1} \frac{P(w_i=1|\mathrm{R},\vec{q})}{P(w_i=1|\bar{\mathrm{R}},\vec{q})} \prod_{w_i=0} \frac{P(w_i=0|\mathrm{R},\vec{q})}{P(w_i=0|\bar{\mathrm{R}},\vec{q})} \tag{18}$$

This formula can be rewritten by substituting notational shorthands: $p_i = P(w_i = 1|\mathrm{R},\vec{q})$: the probability that a term occurs in a relevant document and $q_i = P(w_i = 1|\bar{\mathrm{R}},\vec{q})$: the probability that a term occurs in a non-relevant document and assuming that $p_i = q_i$ for all terms not occurring in query $\vec{q}$. Rewriting and simplifying (18) yields

$$O(\mathrm{R}|\vec{q},\vec{d}) = O(\mathrm{R}|\vec{q}) \prod_{\{t_i \in V_c | w_{di}=1 \wedge w_{qi}=1\}} \frac{p_i(1-q_i)}{q_i(1-p_i)} \prod_{\{t_i \in V_c | w_{qi}=1\}} \frac{1-p_i}{1-q_i} \tag{19}$$

Because in a practical system one is only interested in the (partial) order of documents and not in the absolute probabilities or odds, one can leave out the components in (19)

[22] For the rest of this section we will leave out the document index k to improve readibility.

that are independent of the document. Taking the logarithm of the product (log-odds) yields:

$$\text{RSV} = \sum_{t_i \in V_q} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \tag{20}$$

where V_q denotes the number of unique query terms, so the summation is limited to the terms of the indexing vocabulary that occur in the query. This basic BIR model can only be applied after estimating the parameters p_i and q_i for all query terms, e.g., for each query term we have to estimate the probability that this term occurs in a relevant document (p_i) and in a non-relevant document (q_i).

Robertson & Sparck Jones (1976) discuss four methods to estimate these parameters. The methods start from two different a-priori assumptions concerning term independence and two concerning document ordering.

I1:: The distribution of terms in relevant documents is independent and their distribution in all documents is independent.

I2:: The distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.

O1:: Probable relevance is based only on the presence of search terms in the documents.

O2:: Probable relevance is based on both the presence of search terms in documents and their absence from documents.

All four possible combinations of a term independence assumption (I1 or I2) and ordering principle (O1 or O2) were tested. Combination I2-O2 turned out to be the most effective in practice; it will be discussed below.

Suppose we have a document collection of size N and a query Q. Now p_i and q_i can be estimated for each query term by dividing the document set in four different parts according to the following contingency table: In table 2.4 the total set of documents N

	Relevant	Non-relevant	
$w_i = 1$	r	$n - r$	n
$w_i = 0$	$R - r$	$(N - n) - (R - r)$	$N - n$
	R	$N - R$	N

Table 2.4. Contingency table of term occurrence vs. relevance

can be divided along 2 axes: (i) there are n documents that contain the query term, (ii) there are R documents which are relevant for the query. Finally the number of relevant documents that contains the query term is r.

Robertson/Sparck Jones formula. Suppose now that the distribution of terms is independent both in the set of relevant documents and in the set of irrelevant documents (I2), suppose further that we have full relevance information and that probability of relevance is based on both presence and absence of query terms in documents (O2), then p_i can be estimated by r/R and q_i by $(n-r)/(N-R)$. Substitution into the individual term weight of (20) yields

$$w = \log \frac{r/(R-r)}{(n-r)/((N-n)-(R-r))} \tag{21}$$

This term weighting formula is usually referred to as the *Robertson/Sparck Jones* formula.

Remember however, that probabilistic relevance models presuppose full relevance information. The unmodified formula (20) is undefined if there is no relevance information ($R = r = 0$). Therefore a small constant (0.5) is added to both numerator and denominator of both probability estimates. This is common practice in parameter estimation with incomplete data. Without relevance information (21) can be rewritten as

$$w = \log \frac{N - n + 0.5}{n + 0.5} = w^{(1)} \tag{22}$$

which effectively behaves like a logarithmic inverse collection frequency weight (cf. component t in table 2.2 (Appendix A)). Concluding, the strength of the BIR model is its capability to exploit relevance information. In fact, the BIR model is closely related to the Naive Bayes classifier, which is often used for (supervised) text classification (Lewis, 1998). Without relevance information, the BIR model is very weak in comparison with standard *tf.idf* since it lacks a term frequency component and document length normalization.

2.6.1.2. *2-Poisson distribution based model.* Because the BIR model clearly had its limitations a new probabilistic model based on the 2-Poisson distribution was developed by Robertson & Walker (1994) (cf. section 2.2.5 for a presentation of the 2-Poisson distribution). A first step is to replace the individual term weight in formula (20) by formula (23). Here p_{tf} and q_{tf} are the probabilities that a term occurs with a frequency tf in a relevant or non-relevant document respectively. p_0 and q_0 denote term absence in relevant and non-relevant documents.

$$w = \log \frac{p_{tf}\, q_0}{q_{tf}\, p_0} \tag{23}$$

Combining this new weighting formula (23) with the 2-Poisson formula (5) yields:

$$w = \log \frac{(p'\lambda_1^{tf} e^{-\lambda_1} + (1-p')\lambda_2^{tf} e^{-\lambda_2})(q' e^{-\lambda_1} + (1-q')e^{-\lambda_2})}{(q'\lambda_1^{tf} e^{-\lambda_1} + (1-q')\lambda_2^{tf} e^{-\lambda_2})(p' e^{-\lambda_1} + (1-p')e^{-\lambda_2})} \tag{24}$$

Early experiments with estimating the lambdas directly from the term frequencies yielded poor results. This might be due to the estimation methods as such or to the lack of sufficient data to estimate the multitude of parameters (4 per term). Robertson and Walker experimented with an approximation of formula (24) by a simpler formula which exhibits the characteristics of the orginal model:

(1) The weight is zero for $tf = 0$,
(2) it increases monotonically with tf
(3) but to an asymptotic maximum,
(4) which approximates the Robertson/Sparck Jones weight that would be given to a direct indicator of eliteness.

The *constructed* formula which exhibits these characteristics is simply the multiplication of the original Robertson/Sparck Jones weight (22) with the function $tf/(constant + tf)$:

$$w = \frac{tf}{k_1 + tf}\, w^{(1)} \tag{25}$$

The value of the constant k_1 has to be defined empirically, but is fortunately quite stable across collections.

In summary, to overcome some of the deficiencies of the BIR model, Robertson and Walker have found inspiration in the 2-Poisson model. Because estimation of all parameters of the resulting model is intractable, an approximation is suggested, which multiplies the original weight with an asymptotic scaling function, which scales the weight depending on the raw term frequency.

Robertson and Walker propose a second modification in order to cope with document length differences. Two hypotheses are presented for the relation between document content and document length:

scope hypothesis: A long document consists of a series of unrelated short documents (e.g., a news bulletin).

verbosity hypothesis: A long document is basically an extended version of a short document covering a topic in more detail.

In reality a document collection like the TREC collection probably contains examples of documents which support either one or both hypotheses to some degree. The obvious approach to deal with the scope hypothesis would be to implement procedures for automatic topic segmentation of documents. This approach has gained some response in the TREC community in the form of *passage retrieval* techniques, which however, usually work with fixed window subdocuments. Robertson and Walker have chosen to start from the verbosity hypothesis. Given some additional independence assumptions, this hypothesis leads to a refined model where the weighting is adjusted for documents which have a length which deviates from the average document length. The resulting model leaves term weights unchanged for documents with a length equal to the average document length, but lowers term weights of very long documents and increases term weights of relatively short documents.

We refer to Robertson & Walker (1994) for a formal presentation of the resulting model. We limit ourselves to presenting two formulas which approximate the document length corrected model. The first formula is:

$$w = \frac{tf}{\frac{k_1 \times d}{\Delta} + tf} w^{(1)} \tag{26}$$

where d is the document length: $\sum_{i \in L} tf$ and Δ the average document length: $\sum_{j \in N} \sum_{i \in L} \frac{tf_{ij}}{N}$. This is basically a revised version of (25) which normalizes tf for lengths which differ from the average document length. The second formula defines a correction factor which approximates the behavior of a function which has a maximum for d approaching zero, equals zero for $d = \Delta$ and approaches a minimum for d approaching infinite length. This complex function can be approximated by:

$$\text{correction factor} = k_2 \times |Q| \frac{\Delta - d}{\Delta + d} \tag{27}$$

Again k_2 is a tuning constant which has to be determined empirically. Note that this is a global formula component which is added to the RSV after all term weights have been processed. $|Q|$ represents the number of query terms.

Robertson and Walker finally suggest a within-query term frequency reweighting (tf_q) which has some plausibility but for which the theoretical motivation (especially in combination with the within-document term frequency and document length models) is weak.

$$w = \frac{tf_q s_3}{k_3 + tf_q} w^{(1)} \tag{28}$$

A high value for the tuning constant k_3 has turned out to be optimal, in fact a linear tf_q function was taken by taking $k_3 = \infty$ and $s_3 = k_3 + 1$. Robertson & Walker (1994) report on experiments with the TREC-2 collection with several weighting functions based on various combinations of the term-weighting function components discussed in this section, carried out with the Okapi system at City University, London. In the tradition of the SMART group, the weighting functions have an abbreviated code, starting with BM, 'Best Match'. Table B.1 in appendix B gives an overview of the main Okapi weighting functions. The best known model of this family of Okapi models is BM25:

$$\begin{aligned} \mathrm{RSV}(q, d_j) = \sum_{i=1}^{V_q} \Big(s_1 s_3 \times \frac{{tf_{ij}}^c}{K^c + {tf_{ij}}^c} \times \log \frac{N - n + 0.5}{n + 0.5} \times \frac{tf_{iq}}{k_3 + tf_{iq}} \Big) \\ + k_2 \times |Q| \, \frac{\Delta - d}{\Delta + d} \end{aligned} \tag{29}$$

Often a simplified version is used where $k_2 = 0$, leaving out the global document length correction component, $b = 0.75$ and $k_1 = 2$:

$$\mathrm{RSV}(q, d_j) = \sum_{i=1}^{V_q} tf_{iq} \times \frac{tf_{ij}}{2 \times (0.25 + 0.75 \times \frac{d}{\Delta}) + tf_{ij}} \cdot \log \frac{N - n + 0.5}{n + 0.5} \tag{30}$$

This variant, also known as the Cornell or SMART version of BM25 (Singhal et al., 1995) has been used in some of the experiments reported in chapter 6.

To summarize this section on probabilistic relevance models, the BIR model evolved into the BM25 model via modifications related to document length normalization and term frequency. Especially the term frequency work has been inspired by the 2-Poisson model. However, due to the complexity of the theoretical model, implementations are necessarily based on simplified models. The term weighting formulae modifications were constructed in order to approximate the curve shapes of the theoretical model. The resulting term weighting function has proven to be effective (BM25 has been and still is the preferred model for many TREC participants, cf. (Robertson et al., 2000) for an overview of Okapi performance at TREC) but we think the theoretical motivation is less elegant (in the sense of simplicity) than the recently proposed language model based IR models which we will discuss in section 2.6.3. Another drawback of the BM25 term weighting function is that it contains quite a few tuning constants, which have to be adapted for each new collection.

2.6.2. Inference based models. The second class of probabilistic IR models are the inference based models. The main idea underlying this approach is that IR is a process

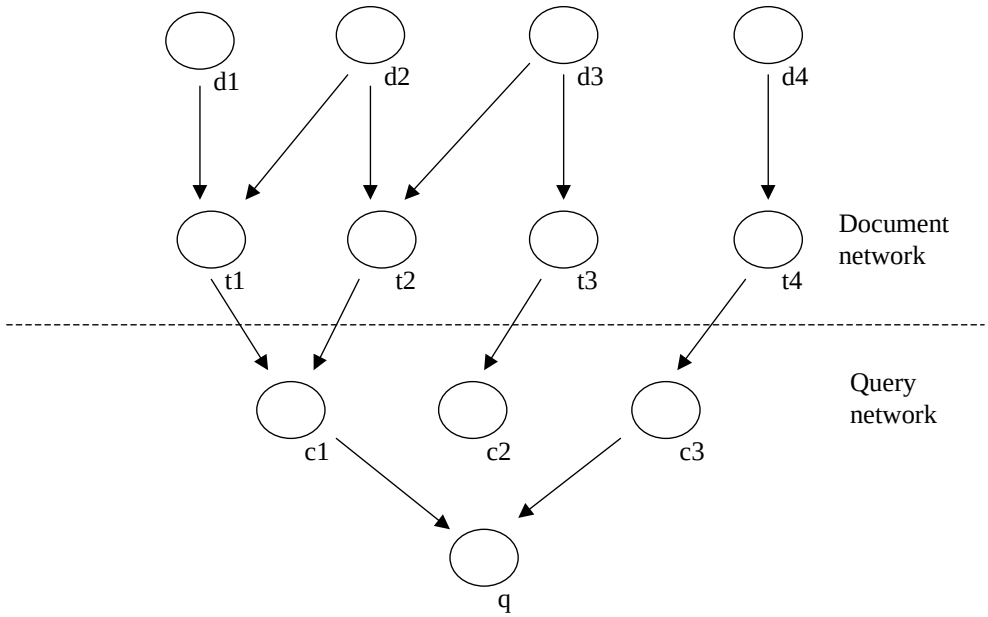

Figure 2.4. example of an inference network

of uncertain inference[23]. These models can be seen a blend between logic and probability theory. An important aspect of this class of models is their extendibility and collection independence. In inference based models it is easy to combine different information sources: evidence is not limited to the query formulation, but can also include knowledge about the user, the domain etc. These parameters are collection independent, whereas the relevance based models contain parameters which have to be adjusted for every new collection. Inference models include two subclasses:

(1) Inference networks. These are Bayesian decision networks.
(2) Probabilistic inference models. These models are based on non-classical logics where the semantics of inference is modeled in probability theory.

2.6.2.1. *Inference network-based retrieval model.* Inference networks are in fact Bayesian networks. A Bayesian network is usually depicted as a directed acyclic graph where nodes represent random variables and arcs denote causal relationships. An inference network for IR consists of two layers, the document layer (which is built off line) and the query layer which is built on-line and can be interactively modified by the user. An example inference network is drawn in figure 2.4. Here the nodes represent random variables which can have the value *true* or *false*. The random variables are associated with observing a certain document ($d_1, \ldots, d_4$), certain index terms ($t_1, \ldots, t_4$), certain query concepts ($q_1, \ldots, q_3$) and a query (q). Nodes in the network are connected by arcs

[23]This approach is based on an epistemological view on probability in contrast with e.g., the frequentist view on probability which underlies the approach taken in 2.6.3. See Manning & Schütze (1999) for some background on the frequentist versus epistemological view on probabilities.

that represent conditional probabilities. Suppose we observe document d_2, then the probability that we observe t_1 or t_2 is defined by the conditional probability on the arcs connecting document and respective term nodes. These conditional probabilities are usually estimated by taking a normalized *tf.idf* weight.

The query and document network are connected with links that connect document and query concepts. Now, in order to compute the belief $P(q = \text{true}|d_i = \text{true})$, first the node corresponding with d_i is instantiated with "true" . Subsequently the probabilities of each node can be iteratively updated, layer by layer (going from parents to children) eventually leading to $P(q = \text{true}|d_i = \text{true})$. The query network can consist of layers of intermediate nodes, which model the evaluation of boolean or weighted sum operators. In fact an inference network can be used as an implementation platform for several classical retrieval models.

A nice property of the inference net framework is that multiple representations (e.g., single terms, phrases, controlled terms) of the same document can be represented in the same network, and also that an information need can be modeled by a parallel evaluation of different queries (Turtle, 1991).

The INQUERY system (Broglio et al., 1995), based on the inference network-based retrieval model, has performed well on the TREC evaluation tasks, clearly showing the feasibility of this approach. A weak point is that the conditional probabilities have to be estimated, while the model does not include a theory internal framework to estimate these probabilities.

2.6.2.2. *Probabilistic inference models.* As figure 2.3 shows, probabilistic inference can be seen as a special case of both probabilistic and logical models. We have chosen to present this class of models under subsection 2.4.5 (under the heading of the class of logical models).

2.6.3. Language models. The recent rise in popularity of corpus based as opposed to knowledge based methods in computational linguistics (see Manning & Schütze, 1999) has produced some interesting cross-fertilizing side effects in IR. In 1998, three new IR models were proposed in independent publications, which were all based on the notion of a statistical language model (LM), which is a standard component in speech recognizers (Jelinek, 1997) or statistical MT techniques (Brown et al., 1993). These three new models (Ponte & Croft, 1998; Hiemstra, 1998; Miller et al., 1999b) were all justified with the argument that it is not necessary to use parametric models for relevance ranking like the 2-Poisson or m-Poisson model, because models can be built from the data themselves. These new models were competitive with the best known retrieval systems at the time[24]. Rather than modeling the probability of relevance they model the probability that a document could be the basis for the user's query i.e. the probability that the query is generated by a statistical language model based on this document. An early application of language models (hidden Markov models) for document ranking is described in (Mittendorf & Schäuble, 1994). Key difference between the more recent proposals and this work is that the recent work estimates the probability of a query *given the document*, whereas in (Mittendorf & Schäuble, 1994) the probability of a text fragment (in a passage retrieval application) is estimated on the basis of the query. There is a lot of evidence

[24]Okapi, SMART and INQUERY (cf. (Voorhees & Harman, 1999a)).

that this new class of models is a productive class. In 1999, two new systems entered the TREC-8 evaluation which investigated new variants and extensions of the LM approach to IR (Ng, 2000a), (Berger & Lafferty, 2000), confirming that this new class of IR models has high potential. Also Kwok (Kwok, 2000) showed that the model developed at CUNY can be shown to be partly based on a LM approach. In the following subsections we will discuss first the Ponte & Croft model, than (in one subsection) the Hiemstra and Miller et al. model and subsequently several variant LM based models that take a slightly different starting point.

2.6.3.1. *The Ponte & Croft model.* The earliest IR model based on language modeling (LM) techniques was published by Ponte & Croft (1998). The leading intuition is that queries are not created without any knowledge of documents, but that users have a reasonable idea which terms occur in relevant documents and will use those terms in a query. The basic idea is now to estimate the probability of a query given a document based language model and use this probability to rank the documents instead of the probability of relevance. The LM-based model presented by Ponte&Croft already contains the main ingredients which we will see in later variations: the probability that a query vector has a certain form is modeled by a multiplication of the probabilities of the individual terms (since the terms are assumed to be independent). These probabilities are estimated from the *local*[25] document model and (if the document does not contain the query term) from the (global) corpus. The Ponte&Croft model differs in two aspects from the Hiemstra and Miller et al. model (which are quite similar). Firstly, the smoothing procedure for parameter estimation is more complex, involving both a collection model based back-off estimator and a factor penalizing sharp deviations from the average probability of occurrence of documents containing the term. Secondly, Ponte&Croft model a query as a binary vector. The model is based on independent estimates of the probability that a term is a member of the query (1) or is not a member (0) of the query. The ranking function is thus a multiplication of N probabilities, N being the number of different index terms.

$$P_{set}(Q|D) = \prod_{t \in Q} P(t|D) \times \prod_{t \notin Q} (1 - P(t|D)) \tag{31}$$

The ranking function also contains a component which models the probability that a term is not generated by the model. The latter component is special feature of the Ponte & Croft model, which captures the idea that a document that discusses a lot of side issues (according to the scope hypothesis, cf. section 2.6.1.2) is probably less relevant than a document that just covers the query topic and thus provides some kind of implicit length normalization. However, from an implementation point of view, this complicates document scoring considerably.

2.6.3.2. *The Hiemstra and Miller et al. models.* Instead of describing a query as a binary vector, one can also treat a query as a sequence of terms $T_1, \ldots, T_n$, which has the advantage that contextual phenomena like phrases can be modeled when appropriate models

[25] We use the term *local* here to indicate that this parameter is estimated on the term distribution of the document of which the relevance is tested. Global refers to collection-wide statistics. Term weighting formulae often contain both ingredients (e.g., *tf.idf*).

(cf. bigram or trigram) are used.

$$P(Q|D) = P(T_1, T_2, \ldots, T_n|D) = \prod_{i=1}^{n} P(T_i|D) \tag{32}$$

In LM terms this can be paraphrased as: the probability of the observation $T_1, T_2, \ldots, T_n$ is equal to the product of the individual term probabilities (in a unigram model).

This query model has been fruitfully applied by several researchers (Hiemstra, 1998) (Miller et al., 1999b). Just like Ponte & Croft, Hiemstra replaces relevance based probabilistic modeling by the query generation metaphor. Miller et al. (1999b) however, ground their LM based approach in a relevance based framework by applying Bayes' rule[26] :

$$P(D \text{ is } R|Q) = \frac{P(Q|D \text{ is } R) \cdot P(D \text{ is } R)}{P(Q)} \tag{33}$$

where $P(Q)$, the prior probability that a query is being posed, is ignored because it is a constant and does not contribute to the ranking function. It is reasonable to assume that the prior probability that a document is relevant is not equal for all documents (cf. the discussion in section 21). BBN experimented with document priors conditioned on document source, document length and average word length. They reported a small improvement (Miller et al., 1999b). We have experimented with a prior conditioned on the document length, which proved quite effective for short queries and with priors conditioned on the form of an URL or the number of inlinks of a webpage (Kraaij et al., 2002). Both information sources yielded considerable improvements in effectiveness in the TREC entry page search task, showing the ease with which external information about documents can be included in Bayesian IR models.

The Hiemstra model and the BBN model are conceptually quite similar. The central probability $P(T_i = t_i|D)$ is modeled by an interpolated (or mixed) language model to compensate for sparse data:

$$P(T_i = t_i|D) = \alpha P_{ml}(T_i = t_i|D) + (1 - \alpha)P_{ml}(T_i = t_i|C) \tag{34}$$

where $P_{ml}(Q|C)$ denotes the language model based on the full corpus capturing the global probability of a term. BBN implements this mixed language model by a two state hidden Markov process. The advantage of using HMM models is that it is easy to extend them with, for example, bigram or synonym models. A disadvantage is that an efficient implementation is not so straightforward.

The approach taken by Hiemstra on the other hand is tailored for implementation using standard scoring algorithms (Frakes & Baeza-Yates, 1992) which are based on processing just the posting lists of the query terms, instead of scoring all the documents in the database. We will describe the Hiemstra model in some more detail, because it is one of the models that we have used in our experiments. Combining (32) and (34) yields:

$$P(T_1, T_2, \ldots, T_n|D) = \prod_{i=1}^{n} \alpha P_{ml}(T_i = t_i|D = d) + (1 - \alpha)P_{ml}(T_i = t_i|C) \tag{35}$$

[26]The idea to express the probability of relevance of a document as a function of the probability that a user uses a certain query term given the assumption that he wants information similar to this document was already formalized in (Maron & Kuhns, 1960).

The maximum likelihood estimates are based on $tf_i / \sum_{i \in D} tf_i$ and $df_i / \sum_m df_i$ respectively, where m is the indexing vocabulary size. A variant global maximum likelihood estimator (mle) applied by Miller et al.: $\sum_k tf_i / \sum_k \sum_{i \in D} tf_i$, where k is the number of documents. The latter estimator is intuitively more straightforward, but the *df* based estimator provides an approximation which is easy to implement (every IR engine maintains a list of *df* values) and which has been used by Hiemstra to give an elegant probabilistic justification for the classical *tf.idf* vector space model (Hiemstra, 1998). We performed experiments with both estimators on several TREC collections, yielding similar retrieval effectiveness (i.e. no significant differences). After taking the log, equation (35) can be reformulated in:

$$\log(P(T_1, T_2, \ldots, T_n|D)) = \sum_{i=1}^{n} \log\Big(\alpha P_{ml}(T_i = t_i|D = d) + (1 - \alpha)P_{ml}(T_i = t_i|C)\Big) \tag{36}$$

which can be transformed into:

$$\begin{aligned} \log(P(T_1, T_2, \ldots, T_n|D)) = &\sum_{i=1}^{n} \log\Big(1 + \frac{\alpha P_{ml}(T_i = t_i|D = d)}{(1 - \alpha)P_{ml}(T_i = t_i|C)}\Big) \\ &+ \sum_{i=1}^{n} \log\Big((1 - \alpha)P_{ml}(T_i = t_i|C)\Big) \end{aligned} \tag{37}$$

Because the second component is a query dependent constant, it can be safely left out[27]. In the most elementary version of the model, α is taken to be a constant. But in principle α could be dependent on the document or even the query and estimated with automatic optimisation procedures like the EM algorithm.

Hiemstra also proposed a basic version of document length normalization. During the ranking process, retrieved documents receive a prior probability

$$\log \frac{\sum_{i \in T_d} tf_i}{\sum_{j \in N} \sum_{i \in T_d} tf_{i,j}} \tag{38}$$

Substituting the estimators in the formula and adding a document prior which benefits longer documents results in:

$$\mathrm{RSV}(Q, D_i) = \sum_{t \in T_q} \left(tf_q(t) \log(1 + \frac{\alpha tf_d(t) \sum_m df(t)}{(1 - \alpha) dl_i df(t)} \right) + \log \frac{dl_i}{\sum_{i \in N} dl_i} \tag{39}$$

2.6.3.3. *LM variant: statistical translation.* Somewhat later than the first three LM publications, a variant model has been proposed where IR is modeled as a statistical translation process (Berger & Lafferty, 1999, 2000). The most important contribution of this model is the integration of synonymy and polysemy into the model. The basic structure of the model is similar to the approach of Miller et al. However, the authors model IR as an instance of statistical translation:

(1) The user has an information need $\mathcal{N}$.
(2) From this need, he *generates* an ideal document fragment $\mathbf{d}_{\mathcal{N}}$.
(3) He selects a set of keywords from $d_{\mathcal{N}}$ and generates a query q from this set. This can be viewed as a *translation* process. Alternatively, this could be seen as an instance of the noisy channel paradigm, where the original document

[27] For applications where score compatibility across queries does not play a role.

> fragment gets corrupted by the communication channel (Shannon & Weaver, 1949).

Now the task for the system is to find those documents which are most likely given the query, i.e. maximize $P(d|q)$, which after applying Bayes' rule amounts to maximizing $P(q|d) \cdot P(d)/P(q)$ analogous to (33). However, instead of modeling $P(q|d)$ directly by a mixed unigram model, Berger and Lafferty integrate work from the IBM statistical MT tradition (Brown et al., 1993, 1990). $P(q|d)$ is modeled by a two step generation process. Firstly terms are sampled from the ideal document, secondly, these terms are translated using a simple statistical MT model (IBM Model 1). The parameters of this model are estimated on a corpus of many query-document combinations, where queries were just sentences taken from the corresponding document. Analogous to the previously discussed approaches, the translation model is smoothed by the background unigram model. The approach has been implemented as the Weaver system. Experiments have been reported on several TREC subcollections, i.e. title and concept queries were constructed from TREC topics 51-100 and were evaluated on the Associated Press and San Jose Mercury News document subcollection. The new model improved average precision with 20-30% wrt. a baseline of a simplified Okapi model, cf. (30). However, the system performed less convincing in the TREC8 ad hoc evaluation: Average precision was 0.2448 vs. 0.2778 for the much simpler Hiemstra model on the title+description run. Efficiency seems to be an issue: "Parameter estimation and document ranking required several days to complete" (Berger & Lafferty, 2000).

2.6.3.4. *LM variant: likelihood ratio.* A different variant model has been proposed in Ng (2000a,b). Instead of estimating the probability of relevance of a document given a query, Ng proposed to use likelihood as an ordering criterion:

> "The idea is that documents that become more likely after the query is specified are probably more useful to the user and should score better and be ranked ahead of those documents whose likelihoods either stay the same or decrease."

In other words, Ng postulates that a document which has a larger likelihood than another document given a certain query is more relevant. More precisely, he proposes the likelihood ratio of a document before and after a query has been posed as a scoring function. "After a query Q is specified by a user, the likelihood of each document changes.". We consider this justification for the use of language models not entirely convincing, since relevance as such is not present in the model, but the other LM based IR models meet similar criticism (Jones & Robertson, 2001).

The ranking function is as follows:

$$\mathrm{LR}(D_i, Q) = \frac{p(D_i|Q)}{p(D_i)} \tag{40}$$

which can be rewritten (Bayes' Rule) as:

$$\mathrm{LR}(D_i, Q) = \frac{p(Q|D_i)}{p(Q)} \tag{41}$$

The probabilities $p(Q|D_i)$ and $p(Q)$ are both modeled by a multinomial distribution. Unlike Hiemstra (1998) and Miller et al. (1999b), Ng models a query as a bag of words

and consequently models the probability of this bag of words, whereas the former authors model the query as an ordered draw, omitting the factorial component from (3). However, the factorial component cancels out, because the model is based on a likelihood *ratio*. Using a linear interpolation approach to compensate for the sparse data, (41) can be converted to a similar ranking function:

$$\text{RSV}(D_i, Q) = \sum_{t \in T_q} \log\left(\frac{\alpha P_{ml}(t|D_i) + (1-\alpha)P_{gt}(t)}{P_{gt}(t)}\right) \tag{42}$$

where $P_{ml}(t|D_i)$ is the usual maximum likelihood estimated on the document based unigram language model and $P_{gt}(t)$ is a good-Turing estimate based on the corpus. The Good-Turing method aims at estimating probabilities of query terms that do not occur in the corpus, by taking away probability mass from the observed terms. Apart from slightly more refined (but also more complex!) parameter estimation procedures, Ng has extended his model with Expectation-Maximization (EM) procedures to estimate the optimal α for each topic and a method to do controlled weighted query expansion based on blind relevance feedback, which is inspired on, but more advanced than the BBN model. The method is both effective[28] and elegant, because it requires no training on previous test collections and the number of tuning parameters is significantly reduced (in comparison with for example BM25 in combination with blind relevance feedback).

It is easy to show that the implementations of the Ng model and the Hiemstra model are almost equivalent:

$$\text{RSV}(D_i, Q) = \sum_{t \in T_q} \log\left(1 + \frac{\alpha P_{ml}(t|D_i)}{(1-\alpha)P_{gt}(t)}\right) + \sum_{t \in T_q} \log(1-\alpha) \tag{43}$$

Comparing (43) with (37), we see that in fact the implementation of the Hiemstra model is equivalent to the Ng model, because omitting the query dependent constant from the ranking function is equivalent to computing a maximum likelihood ratio. However, Ng provides ample empirical proof that taking a likelihood *ratio* instead of the plain probability $P(Q|D)$ ensures cross-query comparability of RSV's. This is due to the term $P(Q)$ in the numerator, which normalizes the widely varying prior probability of Q.

2.6.3.5. *LM variant: cross-entropy reduction.* In this section we present our preferred way of formalizing language models for IR. This particular formalization is also used throughout the chapters 5, 6 and 7. The formalization is a variant of the likelihood ratio models, but models (normalized) document ranking as cross-entropy reduction w.r.t. a background model. This choice is further motivated in chapter 5, but we present it already in this section to show its relationship with the other LM-based IR models. The presentation uses a slightly different syntax, which is presented in table 2.5. The main difference is that the generated index terms are represented by τ, to avoid confusion with the variables t and s for term in target and source term, that we will use in chapter 5. Also the smoothing parameter is represented by λ[29] instead of (1 -α). Starting point

[28] The MIT participation in the ad hoc track of TREC-8, which is based on the LR model, ranked 4th. of 31 groups

[29] This is opposite to (Hiemstra, 2001), where λ denotes the shrinkage parameter.

symbol	explanation
Q	Query has representation $Q = \{T_1, T_2, ..., T_n\}$
D	Document has representation $D = \{T_1, T_2, ..., T_n\}$
τ_i	index term
s_i	term in the source language
t_i	term in the target language
$c(x)$	counts of x

Table 2.5. Common symbols

is equation (41), as usual we assume term independence and take logarithms:

$$\text{LLR}(Q|D) = \log \frac{P(Q|D)}{P(Q|C)} = \sum_{i=1}^{n} c(Q,\tau_i) \log \frac{((1-\lambda)P(\tau_i|D) + \lambda P(\tau_i|C))}{P(\tau_i|C)} \tag{44}$$

In (44), $P(Q|C)$ denotes the generative probability of the query given a language model estimated on a large background corpus C. For each term in the query, the LLR (log likelihood ratio) model judges how much surprise there is to see this term given the document model in comparison with the amount of surprise given the background model. The scores of model (44) depend on the query length, which can be easily normalized by dividing the scores by the query length ($\sum_i c(Q,\tau_i)$). This results in formula (45) for the normalized log likelihood ratio (NLLR) of the query:

$$\text{NLLR}(Q|D) = \sum_{i=1}^{n} \frac{c(Q,\tau_i)}{\sum_i c(Q,\tau_i)} \log \frac{((1-\lambda)P(\tau_i|D) + \lambda P(\tau_i|C))}{P(\tau_i|C)} \tag{45}$$

A next step is to view the normalized query term counts $c(Q,\tau_i)/\sum_i c(Q,\tau_i)$ as maximum likelihood estimates of a probability distribution representing the query $P(\tau_i|Q)$. The NLLR formula can now be reinterpreted as a relationship between the two language models $P(\tau|Q)$, $P(\tau|D)$ normalized by the the third language model $P(\tau|C)$. The model measures how much better than the background model the document model can encode events from the query model; or in information theoretic terms. We prefer to reiniterpret the formula as the difference between two cross-entropies:

$$\text{NLLR}(Q|D) = \sum_{i=1}^{n} P(\tau_i|M_Q) \log \frac{P(\tau_i|M_D)}{P(\tau_i|M_C)} = H(Q,C) - H(Q,D) = CER(Q;C,D) \tag{46}$$

where $H(Q,C)$ and $H(Q,D)$ are cross-entropies and $CER(Q;C,D)$ is a shorthand for this ranking formula that we will call *cross-entropy reduction* in this thesis. Cross-entropy is a measure of our average surprise; so the better a document model 'fits' a query distribution, the higher the score will be.[30] For relevant documents, $H(Q.D)$ will be smaller than $H(Q,C)$, the smaller the cross entropy given the document model is (e.g., when the document language model better fits the observations sampled from the query language model), the higher it will be ranked.

The representation of both the query and a document as samples from a distribution representing respectively the user's request and the document author's "mindset" has several advantages. Traditional IR techniques like query expansion and relevance

[30]The cross-entropy reduction ranking formula can also be reformulated as a difference of two Kullback-Leibler divergences (Ng, 2000a)

feedback can be reinterpreted in an intuitive framework of probability distributions (Lafferty & Zhai, 2001a; Lavrenko & Croft, 2001). The framework also seems suitable for cross language retrieval. We only need to extend the model with a translation function, which relates the probability distribution in one language with the probability distribution function in another language. We will discuss several solutions for this extension in chapter 5. The cross-entropy reduction ranking formula also has a disadvantage: it is less easy to integrate prior information about relevance into the model (Kraaij et al., 2002), which can be done in a straightforward way in formula (32), by simple multiplication.

2.6.3.6. *Comparison of LM-based IR models.* Although this new class of IR models is still under development (cf. chapter 7 for some recent developments) experiments have yielded enough proof that the new framework is quite promising. Although different versions of the model exist, the principal idea: "compute the probability that the query is generated by the document" is shared by all proposals. A weak point of the inital LM based approaches is that none of them explicitly includes relevance in the model. We think that a plausible justification could be to regard $P(Q|D)$ as a probabilistic version of the logical implication. Van Rijsbergen (1986) demonstrated that the retrieval process can be seen as a problem of computing the probability that a document implies the query: $P(D \rightarrow Q)$. Nevertheless, the good performance figures show that the model captures the empirical relevance data quite well. More recent LM-based approaches (e.g., Lavrenko & Croft (2001)) have tried to include relevance in the model (including relevance feedback), we will discuss these models in some more detail in chapter 7.

The Weaver approach is promising from a theoretical point of view, because it integrates polysemy and synonymy into the IR model by explicitly modeling the relationships between terms. Unfortunately this more complex model has not yet proven its value, but it might be too early for a definite judgement. The Hiemstra model gives a probabilistic justification for Salton's classical *tf.idf* vector space model. Besides, the Hiemstra model has been extended for cross-language retrieval in a way which is essentially equivalent to the Weaver model (cf. (Hiemstra, 2001) and chapter 5). The Miller et al. (1999b) model differs from these approaches merely in the implementation aspect, because it uses hidden Markov models. The most important contribution of the Ng model is that it shows that the $P(Q)$ term in the denominator is essential to normalize the RSV over queries, Ng also shows that his model parameters (including query expansion terms) can be optimized locally (i.e. without training on an external test collection), which is important because collection parameters may differ significantly in real life. Our cross-entropy reduction formula is merely a re-interpretation of Ng's model in information theoretic terms. A clear advantage of the re-interpretation is that it shows that ranking is a function of three different language models, which provides a clear avenue for extending the model e.g., for CLIR.

An underestimated problem in LM-based IR is document length normalization. Though different strategies have been applied (Hiemstra, 1998), (Miller et al., 1999a), a convincing approach has not emerged. One explication could be that there is no theory of internal document structure. Documents are considered to be bags of words. In reality, some long documents treat different topics or go on at length about one topic, or a mix. One

could model these assumptions in a mixed model and estimate parameters on a previous collection.

2.7. Conclusions

Whereas manual indexing languages are usually based on pre-coordinated controlled index terms, automatic IR systems take single words from the documents themselves as index terms. Exact match automatic IR systems start from the idea that when a document contains a term, the document is about this term. This clear semantics of index terms can partly explain the popularity of Boolean retrieval systems. However, Boolean systems have their limitations: it is difficult to compose more complex queries and the retrieval result list is often too long or empty. An alternative is formed by ranked retrieval systems, which drop the aboutness assumption and instead take the occurrence of a word in a document as an uncertain indicator for its content. The estimation of this uncertainty has been addressed by a large variety of IR models: logical models, vector space models and probabilistic models. Each of these classes starts from a different intuition to model the semantic space of documents and queries, e.g., a high-dimensional space, a multinomial probability distribution, or a probabilistic inference framework. When we limit our view to ranked retrieval models which have been tested on collections of serious size (TREC is the de-facto standard) there are basically four different types of models that continue to perform well[31]: the vector space model, the relevance-based probabilistic model, the inference network model and more recently the language model-based IR model. All these models have a more or less similar performance. The TREC collection showed some flaws in the original models, e.g., they had to be adapted to deal with heterogeneous document lengths. Some of these models have components of which one could argue that they are tuned to the TREC collection or contain ad-hoc solutions with curve transformations or parameters that have to be trained on prior data. In our opinion the language model-based IR models provide the cleanest solution to the IR problem. The Ng variant does not even need a prior training collection to train its single parameter. The single criticism one could have on the LM-based approaches, is that the model does not include relevance.

In our overview we have focussed on laboratory IR models. It is important to realize that such laboratory IR models are not necessarily good candidates for a real world IR system, for example, a WWW search engine. An important factor of these systems is speed, sometimes quite ad-hoc decisions are taken to maintain a fast response time for these systems (Selberg & Etzioni, 2000). Other aspects like an intuitive interface, integration with translation or query expansion tools, update frequency and robustness, are far more important for the commercial success of an IR service or product than an increase in average precision of 10%. But these aspects are not the topic of our thesis. In this chapter we have presented the state of the art in IR models and techniques. However, our thesis research started in 1994. This has the consequence that the experiments we describe in the following chapters use techniques which are sometimes slightly outdated in comparison with the latest insights. In our experiments we have applied and extended variants of vector space models, probabilistic relevance models (BM25) and

[31] The CUNY model of K.L. Kwok is a hybrid model.

language model based IR models. Our guiding research quastion has been whether linguistic techniques are able to improve purely statistical IR systems. In particular we have applied linguistic knowledge to the IR subproblems of conflation, phrase indexing, synonym expansion and cross-language retrieval. Some of these problems have been studied by applying query expansion and weighting techniques, i.e. without really modifying the core of the IR model. Chapter 4 will show that, naive query expansion (without regarding interactions with the core IR model) does not work. In other words, every study of an IR subproblem benefits from an integrated approach, an approach where the subproblem is studied in the context of the core IR model.

The actual IR systems that have been used in our experiments will be discussed in more detail in chapter 4. This chapter also includes baseline experiments. In the next chapter we will give a brief overview of important IR techniques that are often used to improve the performance of basic IR models.

CHAPTER 3

Compensating for poor queries

In this chapter we will discuss techniques that are frequently applied to improve IR performance, but are often viewed as external to the retrieval models proper discussed in the previous chapter. We will not present an exhaustive overview, but limit ourselves to the techniques used in our experiments. The chapter starts with a discussion of *relevance feedback*, a technique to enhance a query on the basis of relevance information. The feedback information can either be used to re-weight query terms or to expand the query with terms from relevant documents. The latter technique can also be used to improve noisy document representations, e.g., for OCR'ed pages or speech transcripts. Another technique to improve recall is *approximate string matching*, a technique to relate query terms to index terms which are orthographic variants. The chapter concludes with an overview of linguistic techniques for IR and a discussion of stop lists.

3.1. Relevance Feedback

Although automatic indexing techniques have proven to be successful, it became also apparent that these techniques have their limits. In a famous evaluation study on the IBM IR system STAIRS, Blair & Maron showed that the recall of IR systems is often overestimated (Blair & Maron, 1985). One obvious way to improve recall is to expand the query with new terms, a technique usually referred to as *query expansion*. But it is not so obvious for a user to find the right terms to improve recall. A careless selection of new terms can very easily lead to decreased precision. Therefore researchers have tried to develop automatic query modification techniques. These techniques are essentially supervised machine learning techniques (Mitchell, 1996). In machine learning terms, the (supervised) training data consists of relevance information about documents. Suppose that the system knows that a document is relevant, then it could use this data to select new query terms or enhance the weight of query terms which are found in this document. On the other hand, when the system has the information that a document is not relevant, it could decrease the weight of certain query terms. These supervised machine learning techniques for query modification are usually referred to as relevance feedback. The first ideas about relevance feedback and query modification were already published in Maron & Kuhns (1960). Relevance feedback has proven to be a very effective technique which can be explained by the fact that new knowledge is supplied to the system. The query representation of the information need which initially is usually rather short and incomplete can be refined and extended by exploiting the relevance information.

Initially, experiments with relevance feedback were based on explicit supervised feedback by the user. The relevance information can stem from different sources: (i) prior relevance information (ii) relevance information about retrieved documents supplied by the user. In the latter case, retrieval is viewed as an interactive process, which consists of multiple, iterative runs, where the query is enhanced after each retrieval run. The user can provide relevance information concerning retrieved documents by marking documents in the result list as relevant or not relevant.

In the typical ad hoc retrieval scenario, prior relevance data is not available. A situation where a user gives manual feedback is also more difficult to study, because an extra variable is introduced. However, in completely automatic evaluations, it is often still possible to employ a form of relevance feedback. This automatic relevance feedback is based on the assumption that if a document collection contains relevant documents, a state-of the art IR system will rank these documents at the top of the result list. In other words, the probability that a document ranked at the top of the list is relevant is high enough to justify a relevance feedback approach which simply assumes that for instance the top three documents are relevant. This automatic approach is also called *pseudo-relevance feedback*, *local feedback* or *blind relevance feedback*. We will briefly discuss several techniques. A survey of relevance feedback methods can be found in (Harman, 1992).

3.1.1. Rocchio re-ranking. One of the early relevance feedback techniques which is still very influential was pioneered by Rocchio, using the SMART system (Rocchio, 1971). He defined the modified query as:

$$Q_1 = Q_0 + 1/n_1 \sum_{i \in R} D_i - 1/n_2 \sum_{i \in NR} D_i \tag{47}$$

where D_i is a document vector and n_1 and n_2 are the number of relevant and non-relevant documents for which relevance is available. The new query thus consists of a simple addition of the original query plus the scaled relevant document vectors subtracted by the scaled non-relevant document vectors. The technique yielded very good results. In later publications the *Rocchio formula* is also presented as:

$$Q_1 = \alpha Q_0 + \beta/n_1 \sum_{i \in R} D_i - \gamma/n_2 \sum_{i \in NR} D_i \tag{48}$$

where the α, β and γ parameters determine the ratio with which to mix the original query with positive and negative feedback. Often negative feedback is left out, because the effectiveness of negative feedback has not been consistently proved.

3.1.2. Query expansion. One of the reasons why Rocchio's method works so well is that it expands the query with new related terms by adding the document vectors of relevant documents. Query expansion techniques have been studied also independently from relevance feedback. These techniques are based on finding term-term relationships in the document collection. These relationships can be discovered off-line, by using clusteringclustering techniques, or defining a term-term similarity metric. Unfortunately, automatic query expansion based on simple term-term associations without any form of term re-weighting has not shown consistent improvements of retrieval performance (Harman, 1992). This outcome might be due to the simplistic approach used for query

expansion where term-term relationships are not included in the IR model. Recent work (Berger & Lafferty, 2000) shows that query expansion approaches are effective when the term-term relationships are included in the IR model. A second explanation could be the term-term similarity metric itself. Experiments with similarity thesauri have shown a marked improvement of average precision for short queries (cf. Section 2.5.5).

However, most research on query modification has focused on using query expansion and query term re-weighting. Apart from the already mentioned Rocchio method, which has been developed for the vector space model, a large number of variants exist. We will discuss a few.

3.1.3. Local context analysis. *Local context analysis (LCA)* is a feedback technique which has been developed at the University of Massachusetts and has been a successful component of the INQUERY system (Croft & Xu, 1995). The main difference with standard feedback approaches is the use of *passages*, small, fixed size text windows. In local context analysis, the objective is to expand the query, using the results of a first retrieval pass:

(1) Retrieve the top n ranked passages (text window of e.g., 300 words).
(2) Extract noun phrases from these passages. Rank these according to formula (49). The formula especially boosts infrequent concepts which co-occur frequently (*af*) with infrequent query terms.
(3) The top 70 terms of this ranked list are used for query expansion using a linear diminishing weighting scheme.

$$bel(Q,c) = \prod_{t_i \in Q} (\delta + \log(af_{c,t_i})\, idf_c / \log(n))^{idf_i} \tag{49}$$

where af_{c,t_i} is the summation over passages of the product of the term frequency in the query and in the passage, idf_c and idf_i are scaled inverse document frequencies of the concept and query term t_i, based on the passage collection.

Table 3.1 displays some quantitative results to show the power of this approach: local feedback improves upon the basic INQUERY system with a substantial 11%, but local context analysis improves the baseline with a solid 23.5%. LCA shows that a more

system	baseline	local feedback	local context analysis
average precision	0.252	0.279	0.311

Table 3.1. Evaluation of Local Context Analysis on the TREC-4 collection

constrained approach to co-occurrence analysis is useful. Noun groups appear to be good expansion terms.

3.1.4. Blind relevance feedback. Since the original Okapi model does model relevance feedback but does not allow query expansion, a new query expansion technique called *blind relevance feedback* was developed at Cambridge University (Jourlin et al., 1999). The essence of the technique is the computation of a so-called "offer weight" for every term in the top R documents.

$$ow(t_i) = r \cdot log \frac{(r+0.5)(N-n-R+r+0.5)}{(m-r+0.5)(R-r+0.5)} \tag{50}$$

where R is the number of assumed relevant documents, r is the number of documents which are assumed relevant and contain term t_i , n is the total number of documents containing t_i and N the total number of documents. The offer weight formula slightly differs from the Robertson-Sparck Jones formula (21), because the r component has moved outside the logarithm. There is no theoretical motivation for this change (Jourlin, 2000). For query expansion, simply the top T terms are added to the query, weighted according to their offer weight. This will boost especially those terms that occur frequently in relevant documents.

3.1.5. Collection enrichment. Since TREC-6 some groups (City university, University of Massachusetts and AT&T) have explored techniques to use a secondary large corpus for query expansion. This technique is sometimes called *collection enrichment* or more recently *parallel blind relevance feedback*. Unlike normal automatic relevance feedback techniques, this form of feedback consists of an initial retrieval pass on a secondary corpus. The top N documents of this pass are used to modify the query, which in turn is used for the second (final) retrieval run on the target collection. Of course, care has to be taken to ensure that the secondary corpus at least partially covers the same domain as the target collection. A second potential problem is *topic drift*, i.e. the effect that the first retrieval pass will stress different aspects in the topic than the ones that were intended during topic development. The topic developers use only the target collection for the topic creation. So for TREC there is prior knowledge that the target collection contains a minimum number of relevant documents. One way to overcome topic drift is to restrict the first feedback pass to only modify term weights and not to add query terms which are only suggested by the secondary corpus, a technique called conservative collection enrichment (Singhal et al., 1999). Evaluation on the TREC-6, TREC-7 and TREC-8 test collections has shown that (conservative) collection enrichment yields a consistent improvement over “standard” expansion techniques based on the target collection alone. Systems employing collection enrichment perform in fact as the top systems in TREC.

3.1.6. Document expansion. Recent work on spoken document retrieval has shown a novel method to increase recall: *document expansion* (Singhal et al., 1999; Singhal & Pereira, 1999). The problem with retrieval of spoken documents is that, given typical word error rates of 30-40%, a lot of the content words which have been spoken, are actually not recognized. Some speech recognizers give the option to provide an N-best output. Document expansion on a secondary textual corpus can help to enrich the rather poor 1-best transcripts in the following way: the 1-best transcript is used as a query on a large text collection. The top N similar documents are used to select the most relevant terms. Now we add those terms to the 1-best transcript which are also present in the N-best lattice. In a way, this expansion method functions as a form of corpus-based disambiguation. Recently it has been shown that document expansion is also effective on 1-best ASR transcriptions and even perfect text (human transcripts)(Singhal & Pereira, 1999). For short queries the average precision on the TREC-8 SDR test collection improved with 23%, for long queries the improvement was only 3.5%. Improvements are even larger for the corrupted transcripts resulting from speech recognizers. Document expansion was shown to be most effective for transcripts with the highest error rates. All experiments have shown that the secondary corpus which is used for the document

expansion must overlap with the target collection in order to reach these remarkable results.

3.1.7. Conclusion. Concluding we can say, that using standard IR techniques as a first pass to enrich poor data has shown to be very effective and also easy to implement. Experiments have shown that query expansion via blind or pseudo-relevance feedback is superior over techniques that are based on off-line term-term correlation computations like clustering. An exception is query expansion based on similarity thesauri. Best results have been obtained with local context analysis and local feedback and collection enrichment. Interestingly, results can be improved by combining techniques, for example local feedback and expansion based on a similarity thesaurus. Document expansion is effective in case of short queries and noisy data, e.g., audio transcripts.

3.2. Approximate string matching

Another technique to improve recall of IR systems is to allow a "loose" match between query and index terms. Such an approach is especially useful when dealing with corrupted data, e.g., text not corrected for spelling errors or output from *Optical Character Recognition* (OCR). This kind of data is more abundant than one would expect. Careful analysis of the document collection used in the UPLIFT project (a collection of Dutch newspaper articles, see Appendix B, Section C for details), revealed that of a subset of approximately 50,000 unique word forms ± 20,000 were not included in the Dutch CELEX[1] dictionary. We examined a random sample of ± 2,500 of these words to establish why they were not in the dictionary, 10 % of the unknown words turned out to be spelling mistakes. The results of this analysis are summarized in Table 3.2. Indexes based on

46%	proper names
37%	compounds
10%	spelling mistakes
3%	other language
3%	morphological variant not in CELEX
1%	stem (and variants) not in CELEX

Table 3.2. Analysis of a sample of 2500 OOV-words from the UPLIFT corpus

electronic corpora and also user queries will thus often be "polluted" with non-standard spelling and typos. An IR engine is typically not robust with respect to typos, spelling variants or missing diacritics, though sometimes *wildcard* operators are provided to provide a crude substring matching device. Approximate string matching techniques, which are sometimes to referred to as *fuzzy [string] matching*, can help to solve this problem. The technique is indeed related to the fuzzy IR models discussed in Section 2.4.4.1. The main difference is that fuzzy matching concerns robust string matching on the basis of a set of n-grams representation, whereas fuzzy IR models concern robust document matching based on sets of index terms. In the following paragraphs, we will discuss two

[1] CELEX is a lexical database containing Dutch word forms.

techniques for approximate string matching namely the *Levenshtein edit distance* metric and n-gram representations. In Section 6.2.3 we will describe experiments that apply approximate string matching for conflation.

3.2.1. Levenshtein edit distance. The Levenshtein distance metric (Levenshtein, 1966) is based on the number of edit operations that have to be performed on a word A to convert it into word B. The base form of this metric gives equal weight for different edit operations (insertion, deletions), though variations are possible. The primary disadvantage of the Levenshtein method is its inefficiency, the algorithm is usually implemented by dynamic programming methods, which lack the speedup of pre-computed indices. An easy work-around to speed up the retrieval process of morphological variants is to index the target words on character bi- or trigrams. Such an index can be used as a coarse filter. This technique is discussed in the following subsection

3.2.2. Character n-gram techniques. Because implementations of the Levenshtein edit distance metric are inefficient, most fuzzy string matching algorithms are based on character n-gram techniques(de Heer, 1979; Mittendorf, 1998; Kantor & Voorhees, 1997). Character n-grams are (sub)strings of words with length equal to n. An n-gram representation of a word can be produced by extracting substrings of length n at each position within the word. A word with length l is represented by a set of $l - n + 1$ n-grams.

Suppose we have a query term which is not listed in the lexicon and want to find the most similar words (like in a spelling checker application). One solution is to index all words in the lexicon with character trigrams as indexing features. The indexing vocabulary consists then of all trigrams observed in the dictionary. The advantage of such an index is that matching functions can be implemented more efficient. The words in the lexicon are thus converted to a set of (indexed) trigrams. For example, we could represent the inflections of the verb 'to walk' as follows:

walk: $\Rightarrow \{wal, alk\}$
walks: $\Rightarrow \{wal, alk, lks\}$
walking: $\Rightarrow \{wal, alk, lki, kin, ing\}$
walked: $\Rightarrow \{wal, alk, lke, ked\}$

Given the set representation of query and target words, we can define a matching function based on set operations. Table 3.3 shows some set-based similarity measures taken from Manning & Schütze (1999): Now we can compute similarities, e.g.,

$$Dice(walks, walking) = 1/2$$

$$Jaccard(walks, walking) = 1/3$$

The use of n-grams for fuzzy string matching has become quite popular in the last twenty years. Implementations differ though in the type of n-gram representation employed and in the kind of weighting algorithm used in the similarity metric. The following properties can be used to classify n-gram matching methods:

character window size: This is typically 2,3 or 4. A larger size of n means more discriminatory power, but smaller fuzziness. If one character is misspelled, typically n n-grams are incorrect, which means that for short words, longer

SIMILARITY MEASURE	DEFINITION
matching coefficient	$X \cap Y$
Dice coefficient	$\frac{2\|X \cap Y\|}{\|X\|+\|Y\|}$
Jaccard coefficient	$\frac{\|X \cap Y\|}{\|X \cup Y\|}$
overlap coefficient	$\frac{\|X \cap X\|}{\min(\|X\|,\|Y\|)}$
cosine	$\frac{\|X \cap X\|}{\sqrt{\|X\| \times \|Y\|}}$

Table 3.3. Similarity measures for sets of n-grams.

n-grams like quad-grams are not useful, because target word variants will not match.

number of context characters: Usually a word is padded with some leading or trailing blanks to produce a better representation of the fact that a word starts or ends with a certain letter or bigram. In some cases n-grams even cross word boundaries, which can be useful to account for a little bit of context, word order information, e.g., for language identification.[2]

position information: In certain circumstances (e.g., for automatic correction of spelling errors) it can be useful to enrich the n-gram representation with positional (i.e. character order) information. With this information it is easier to reproduce the exact form which formed the basis of the n-gram representation.

similarity metric & weighting function: As in keyword-based retrieval models, the choice of weighting function and similarity measure has a crucial impact on performance. N-gram-based retrieval or matching models have been less thoroughly developed. Apart from the binary vector similarity functions in Table 3.3 which are based on set operations it is also possible to exploit global occurrence statistics about n-grams and or to incorporate statistical data about common spelling errors (insertions, deletions, transpositions) and/or the OCR character confusion matrix.

There are two character n-gram techniques of particular interest for this thesis. The first method (ISM) has been used in our experiments on expanding query terms with morphological variants (cf. chapter 6). The method employs a secondary n-gram index built on the list of document index terms. So, the documents are indexed by index terms (wordforms or stems), which are in turn indexed by n-grams. The retrieval process first expands the query with morphological variants using the n-gram index and retrieves documents using the expanded query. The second n-gram technique has been

[2]Word boundary crossing n-grams were used in the TREC experiments carried out at Johns Hopkins University (Mayfield & McNamee, 1999) cf. Section 2.1.2.

promoted as a language independent method for dealing with morphological normalization. Here, both query and documents are indexed by n-grams instead of full wordforms. The n-gram representation enables matching of morphological variant terms at the cost of some false matches and does not require language dependent techniques for morphological normalization. We will discuss both techniques in some detail in the following two paragraphs.

ISM: Informatie Sporen Methode. An example of a fuzzy string matching architecture which uses a weighted similarity function is ISM, an acronym for "Informatie Sporen Methode" (Information Trace Method) (de Heer, 1979), which was developed in the late 1970s at TNO's institute for Mathematics and Statistics. The ISM system provides robust access to short text records like MARC[3] records, names in a telephone directory or abstracts in different languages using character trigrams padded with one context character. Trigrams do not cross word boundaries. The ranking algorithm contains a salience function which uses global trigram statistics and position information. We have applied ISM for several fuzzy term conflation experiments: as a simple method for morphological normalization (cf. section 6.2.3) and also for robust matching of spelling variants in several European languages (Hiemstra & Kraaij, 1999; Kraaij et al., 2000).

n-grams for document indexing. Recently, character n-grams have been applied as an alternative representation for document indexing. The idea is that the need for morphological normalization can be by-passed by using character n-grams as document index terms instead of full wordforms or stems. The sub-word representation will allow partial word-matches and thus provides a robust matching algorithm, which is practical in situations with many typographical errors like OCR'ed text.

Preliminary experiments were carried out at TREC-2 and TREC-3 by Cavnar (Cavnar, 1995), who used 4-grams. These experiments yielded a disappointing performance level since the system employed outdated term-weighting algorithms. The full potential of using n-grams for document indexing was demonstrated by a series of experiments by Mayfield and McNamee (Mayfield & McNamee, 1999; McNamee & Mayfield, 2001). The main result of these experiments was that for English, character 6-grams perform just as well as full wordforms, but that a combination of both methods gets a relative performance improvement of about 8%. Mayfield and McNamee have also demonstrated that the n-gram technique is quite attractive and competitive for a multi-lingual system. The n-gramming approach seems especially attractive for compounding languages. Matching compounds with compound constituents is a requirement for effective retrieval for compounding languages (Pohlmann & Kraaij, 1997a; Braschler & Ripplinger, 2003). Many compound splitting procedures require language dependent components such as a lexicon or morphological production rules (Vosse, 1994). N-gram indexing enables matching of compounds with compound constituents without any external language dependent component. Recently, many other researchers have experimented with n-gram indexing for European languages. An interesting recent empirical study is Hollink et al. (2003), which focuses especially on combination techniques and presents experiments with eight

[3]MAchine Readable Cataloguing: a standard format for the description of the bibliographical data of a publication.

European languages. N-gram indexing and matching is shown to be an effective supplementary technique (in combination with full wordform indexing), although not giving consistent improvements for all eight languages.

3.3. NLP FOR IR

Natural language processing (NLP) is the branch of computational linguistics which is concerned with building models and tools that process human language. For many years, NLP was more or less synonymous with rule-based approaches and symbolic representations that were rooted in theoretical linguistics. Theoretical linguistics is concerned with describing (and explaining) expressions of natural language using a rule-based symbolic framework. Traditionally several levels of linguistic analysis are distinguished. The relevant levels for (written) documents are:

morphology: is concerned with assigning an internal structure to words.

syntax: is concerned with assigning an internal structure to sentences in terms of grammatical relationships.

semantics: is concerned with interpreting the meaning of a sentence in terms of an unambiguous formal language.

discourse analysis: is concerned with the analysis of language phenomena that exceed the sentence level e.g., referring expressions.

The theoretically motivated rule-based approach has several limitations for practical applications: (i) real-life expressions of natural language (i.e. written documents or spoken text) are often not well-formed resulting in rejection by the analysis module, (ii) it is very time-consuming to compile a set of rules that describes all and only well-formed expressions belonging to a specific fragment of natural language, (iii) it is very difficult to manually construct rule-sets for the disambiguation of multiple analyses, while ambiguity is present at all levels of linguistic analysis.

An alternative to hand-crafting rule-sets for symbolic analysis of natural language is to use statistics and hand-annotated data in order to train models that can analyze natural language. This data-driven method is usually called the *corpus-based* approach. Corpus-based methods have become increasingly popular since the end of the 1980s and overcome the disadvantages of symbolic approaches to a large extent. Corpus-based methods are especially effective for lower levels of linguistic analysis (e.g., morphological analysis, POS-tagging and chunking). For a more comprehensive overview of natural language processing we refer to Jurafsky & Martin (2000). A good introduction into corpus-based methods for NLP can be found in Manning & Schütze (1999).

The distinction between rule-based and corpus-based approaches is particularly relevant for our thesis since we are interested in embedding linguistic resources (which are usually rule-based) in statistical IR systems. There is a long tradition of research aiming at the improvement of document retrieval systems through the application of linguistic knowledge. The intuition is that since document retrieval deals with text, insights from linguistics and natural language processing must have added value over pure statistical systems. However, the application of linguistic methods in IR has resulted in rather modest performance improvements (with the exception of question-answering systems, of which the discussion falls beyond the scope of this thesis). It has proven difficult to

improve upon purely statistical baseline systems that lack a detailed symbolic analysis of language (Sparck Jones, 1999). However, morphological normalization of index terms for languages with a rich morphology turned out to be fruitful. This is maybe not surprising, since most IR models use words as central units for the representation of documents; any higher level structure (sentence level, or document level) is usually ignored. Other levels of linguistic analysis that have been investigated in the context of document retrieval are syntax and (word) meaning. Syntactic analysis can be used for disambiguation in cases of part-of-speech (POS) ambiguity (e.g., the Dutch word 'kussen' can be a verb or a noun meaning either *to kiss* or *pillow* respectively) or the recognition of complex index terms (phrases). In IR, linguistic analysis at the level of meaning has been mostly restricted to lexical semantics, examples are attempts to use abstract semantic concepts for indexing or the use of synonyms for query expansion. We will discuss the application of these linguistic techniques for document retrieval in the following subsections.

3.3.1. Morphological normalization: stemming, lemmatization. One of the techniques employed in Information Retrieval (IR) to improve effectiveness is normalization of document and query terms. By reducing morphological variance of terms e.g., by mapping singular and plural forms of the same word on a single base form (stem), the query-document matching process can be improved. The normalization process generates so-called conflation classes. Members of conflation classes are treated as if they were equivalent terms. In practice, this means that during document indexing and query analysis, full wordforms are replaced by an index term representing the conflation class. This is usually the normalized form to which all members can be reduced, but it is not necessarily a well-formed word since it just acts as a placeholder for the class. Morphological normalization is usually called *stemming* in an IR context. Sometimes the term *lemmatization* is used, which is restricted to approaches that produce *lemmas* as base-forms. There are two main approaches to achieve morphological normalization. One could either attempt to reduce affixes (usually suffixes) by simple substring removal operations or even truncation. These simple methods usually do not produce morphologically well-formed base-forms. A more principled approach is to apply morphological analysis grounded in linguistic theory about word formation. This method does produce well-formed base-forms which is important in case of showing feedback terms to the user or to access translation dictionaries in the case of a cross-language setting. In addition, such a knowledge rich approach will have a correct coverage of irregular morphology. Three morphological phenomena are of particular interest to IR: inflection, derivation and compounding. The aim of normalization is to group morphological variants that have a similar meaning. Normalizing inflectional variants is usually a meaning-neutral operation. However, the semantic relationship between derivational variants can range from very close to quite distinct e.g., *like, likely* , *art, artist* or *unite, union*. Compound analysis (also called *decompounding* or *compound splitting*) is an additional normalization technique for Germanic languages, since these have a productive compounding capacity. This means that new words can be formed by concatenating existing words. Decomposition of these compound words into their constituting morphological base forms is important for IR, since these compounds can usually be paraphrased by a noun-phrase construction, e.g., "vliegangst" and "angst om te vliegen" (*fear of flying*). Normalization

of compounds will enable a match between both forms of the same composite concept and partial matches with related words after compound splitting, e.g., 'luchtvervuiling' will match with 'vervuiling' Several algorithms have been proposed for compound splitting.. They either use a lexicon (e.g. Vosse, 1994) or a corpus (e.g. Hollink et al., 2003) as a resource for the identification of candidate base forms which can form compounds. We will discuss the results of several comparative studies concerning stemming algorithms in the rest of this section.

Harman (1991) compared three well-known stemming algorithms for English: the S–stemmer, the Lovins stemmer (Lovins, 1968) and the Porter stemmer (Porter, 1980). Harman contrasted these suffixing algorithms with a baseline of no stemming at all. After a detailed evaluation[4], Harman reached the conclusion that none of the stemming algorithms consistently improve performance. The number of queries that benefit from the use of a stemmer is about the same as the number of queries that deteriorate.

Popovič & Willett (1992) investigated whether suffix stripping would be effective for a morphologically more complex language like Slovene. They developed a Porter-like algorithm for the Slovene language and tested this algorithm on a small Slovene test collection[5]. Their experiment shows a significant improvement in precision (at fixed retrieval of the 10 most highly ranked documents). Popovič and Willet's study also included an interesting control experiment. The Slovene test corpus was translated to English and the experiment was repeated. The results of this control experiment confirmed Harman's conclusion that Porter-like stemming does not improve retrieval for English documents. This suggests that the effectiveness of stemming is strongly related to the morphological complexity of a language.

Krovetz (1993) investigated whether more linguistically motivated stemming algorithms would be effective for English and compared them with the Porter algorithm. Krovetz evaluated the performance of four different stemming algorithms using standard test corpora for English (CACM, TIME, NPL and WEST): Porter, revised Porter (a dictionary is used to check whether the resulting stem really exists), an inflectional stemmer and a derivational stemmer (removes both inflectional and derivational affixes).

Surprisingly, Krovetz found that all stemmers yield a significant improvement[6] over no stemming. The derivational stemmer generally gave the best results. Krovetz noted that improvements due to stemming increase at higher levels of Recall and that derivational morphology is responsible for improvement at high levels of Precision. Document length also seems to be of importance; the best results are obtained with short documents (CACM and NPL collections). It is interesting to note that although both Harman and Krovetz have evaluated the Porter algorithm using the same test collection (CACM) and (almost) the same evaluation measure (AP[0.20,0.50,0.80] vs. AP[0.25,0.50,0.75]), they do not reach the same conclusion. Harman concluded that Porter does not yield a statistically significant improvement over a baseline without stemming whereas Krovetz found that there is a significant improvement.

---

[4]Evaluation measures used were: average precision at 0.20, 0.50 and 0.80 recall, van Rijsbergen's E-measure, number of queries that fail (i.e. 0 recall) at 10/30 documents retrieved and total relevant retrieved at 10/30 documents retrieved

[5]The test collection consisted of approximately 500 documents and 48 queries.

[6]Figures range from from 1.3 to 45.3% improvement in average Precision at Recall 0.25, 0.50 and 0.75.

Hull (1996) argued that current evaluation measures such as average Precision and average Recall are not ideally suited for evaluation of retrieval techniques in general and stemming strategies in particular. Hull claimed that average performance figures need to be validated with careful statistical analysis and that detailed analysis of individual queries can uncover important differences that are not found using the traditional measures. Besides the standard average precision at 11 recall points (0.0,0.10,...1.0) (APR11) which he used for comparison with other results, he proposed two new evaluation measures, average precision at 5-15 documents examined (AP[5-15]) and average recall at 50,60,...150 documents examined (AR[50-150]), which he claimed are more suited to estimate performance for shallow searches and more in-depth searches respectively. He subsequently adapted these measures to normalize for query variance by averaging over within-query rank or score. Using these measures, he evaluated the performance of five different stemming algorithms (remove-s, Lovins, Porter, Xerox inflectional stemmer, Xerox derivational stemmer) using the TREC test collection (Harman, 1995). Statistical tests have been applied and detailed, per-query analysis were carried out to identify probable causes for differences between stemmers. Hull concluded that stemming in general is almost always beneficial, except for long queries (i.e. full TREC queries) at low Recall levels, but he was unable to demonstrate significant differences between suffix stripping algorithms like Porter and Lovins and the linguistic stemming algorithms.

Experiments with Dutch, German, Finnish and Swedish compound splitting yielded significant improvement in retrieval performance on several test collections (Kraaij & Pohlmann, 1996b; Sheridan & Ballerini, 1996; Pohlmann & Kraaij, 1997b; Braschler et al., 2002; Braschler & Ripplinger, 2003; Hollink et al., 2003).

It is not possible to compare different approaches to morphological normalization across languages by looking at differences in mean average precision, since most techniques (except n-gram indexing) are language dependent and quantitative comparisons across collections cannot be made. But overall, it seems that normalization usually improves the mean average precision of a system. As one would expect, normalization is more effective for languages with a rich morphological variation like German or Finnish.

3.3.2. Phrase indexing. Phrase indexing is a technique to extract complex index terms which has the goal to construct more precise content descriptions of documents and queries. Whereas morphological normalization is mostly aimed at enhancing recall, the principal goal of phrase indexing is improving precision. The idea of phrase indexing is that phrases are less ambiguous and more precise than index terms consisting of a single word. The word *mug* is ambiguous, but the phrases *coffee mug* and *mug shot* are not ambiguous, and the phrase *air pollution* is more specific than either *air* or *pollution*. The intuition is clear: phrases help to build unambiguous index terms and can be used to enhance precision of a retrieval system by using more specific index terms. The fact that phrases are also an effective instrument for IR is proved by its widespread use for Web search.

Phrase indexing has been studied by many authors. There are two main approaches to phrase indexing:

statistical phrases: A phrase is usually defined as two contiguous non-stop words that occur at least X times in a corpus

syntactic phrases: A phrase is defined as a complex syntactic constituent, usually in the form of a noun phrase.

Statistical phrase indexing is relatively simple: phrases can be identified by scanning the document collection using a stop list. In principle, statistical phrases could be longer than just bigrams, but longer phrases would lead to highly specific index terms, which are less useful for search. Also, allowing very specific index terms makes the approach less scalable, since the index size would grow very rapidly if also sub-terms would be indexed. Statistical phrases have been a standard component in the TREC experiments with e.g., SMART (Mitra et al., 1997) and INQUERY (Allan et al., 1996).

Syntactic analysis is a process that involves much more knowledge, although noun-phrases can be determined in a relatively light-weight process involving just POS-tagging and shallow parsing. The Twenty-One system (ter Stal et al., 1998) is an example of a system that uses maximal noun-phrases as index terms. Most other researchers reduced noun-phrases to a set of term pairs (Strzalkowski et al., 1997; Jacquemin & Tzoukermann, 1999; Zhai et al., 1997). Several researchers worked on phrase indexing for Dutch (ter Stal, 1996; Pohlmann & Kraaij, 1997a). Kraaij & Pohlmann modeled their approach after the work of Strzalkowski, who extracted head-modifier pairs from complex noun-phrases. The syntactic structure of a complex noun-phrase can help to exclude irrelevant term-pairs, e.g., *relational manager* is not a relevant term pair for *relational database system manager*. However the internal structure of complex noun phrases cannot always be determined by looking at the syntactic categories of the individual words. Therefore, heuristic rules (like right branching) may have to be applied. Experiments with the Dutch UPLIFT test collection have shown that syntactic phrases have potential to improve retrieval performance (Pohlmann & Kraaij, 1997a). A comparison with statistical phrases showed that statistical phrases can yield a similar performance gain, once compounds are split (Kraaij & Pohlmann, 1998). It is not clear though whether these gains would still be significant with respect to a state-of-the-art baseline system like Okapi or a generative probabilistic model. Buckley found that most of the gains from phrases disappeared when more sophisticated term-weighting techniques had been developed (Mitra et al., 1997).

3.3.3. Word meaning. There are several phenomena, usually regarded as part of lexical semantics, that have a significant impact on IR performance. Lexical semantics is concerned with word meaning and the relationship of word meaning to sentence meaning. The following meaning relationships are especially relevant for IR:

synonymy: A single concept/meaning is conveyed by different words.
homonymy: A single word can have several unrelated meanings

These phenomena are important since many IR models implicitly assume a one-to-one relationship between words (stems) and concepts (meanings). The common phenomena of synonymy and homonymy make clear that this assumption does not hold and that the relation between a document and its meaning is characterized by uncertainty (cf. chapter 1). In the next two paragraphs, we will discuss methods that are specifically aimed at addressing synonymy or homonymy in order to improve retrieval effectiveness.

Leveraging synonymy relations for IR. Dealing with synonymy and the more general phenomenon of paraphrases is one of the core challenges for IR systems, since it is rule

rather than exception that concepts of interest are described using different terms by different authors. An ordinary user often does not know all relevant terms which are used. Expert users are well aware of the synonymy problem and produce so-called faceted queries (Pirkola et al., 1999), i.e. queries that consist of a conjunction of concepts. Each concept in turn is represented by a disjunction of synonyms. Although such structured expert queries can be quite effective, they are seldom used by naive users due to the complexity of the query language.

Several authors have investigated whether query expansion with synonym terms retrieved from a thesaurus can improve retrieval performance and in particular recall. Automatic query expansion using pseudo-feedback methods has shown to work quite well (cf. section 3.1). Automatic query expansion with synonyms is more problematic since the sense ambiguity of query terms might lead to expansion with irrelevant synonyms. We will discuss two experiments, with query expansion based on Wordnet.

Voorhees (1994) carried out an experiment with manual query expansion using synonyms taken from Wordnet (Miller, 1990). Wordnet is a lexical database of the English language. Synonymous lemmas are organized in so-called *synsets*, each representing a single concept. Voorhees' experiments were aimed at finding an upper bound of the performance of a retrieval system augmented with query expansion. She manually determined which terms should be expanded and resolved any sense ambiguity. The weights of synonyms in the expanded query were *normalized* in order to keep the relative weights of original query terms unaffected. Query expansion did improve retrieval performance for short queries, showing the potential of the technique.

Hand crafted thesauri such as Wordnet are often not specific enough for query expansion and frequently lack specific query terms such as proper nouns and technical terms. Co-occurrence-based thesauri such as the ones discussed in Qiu (1995) and Jing & Croft (1994) are constructed by identifying terms that frequently co-occur in text window. These kind of thesauri can capture domain-specific senses since they are based on a corpus, but might miss important synonyms, since for example the words *astronaut* and *cosmonaut* are seldom used in the same document. A third class of synonyms can be constructed from lists of head-modifier pairs. Words that occur in similar noun phrase contexts have an increased probability of being related. Mandala et al. found that the corpus-based thesaurial expansion was more effective than expansion based on Wordnet. A combination of the three thesauri proved very effective, especially for short or medium length queries. Note that sense disambiguation was done implicitly, by weighting expansion terms by the weighted similarity of the term with each of the query terms, thereby favouring expansion terms that are compatible with the complete query.

Concluding, query expansion based on synonym relations encoded in hand-crafted thesauri can indeed improve retrieval performance of short queries, although the effect is not as large as query expansion based on thesauri that have been automatically generated from the document corpus.

Word sense disambiguation for IR. Since many words have multiple senses, queries will often retrieve irrelevant documents. This problem could be overcome if we could disambiguate the query terms and restrict retrieval to documents containing query terms with the correct senses. A theoretically even more attractive solution would be to index documents on concepts instead of disambiguated words. One could e.g., use synset

numbers from Wordnet as index terms. Such an approach would require automatic word-sense-disambiguation (WSD) for words in the documents and in the query. Even cross-language search can be supported when WSD methods are available for both target and source language and the conceptual language is language independent (Ruiz et al., 2000). Conceptual indexing can be seen as a knowledge-based equivalent to techniques for dimensionality reduction that also aim at reducing homonymy and grouping synonyms (Deerwester et al., 1990; Hofmann, 1999).

Word sense disambiguation is a problem which has been studied for decades, both in the field of computational linguistics as well as in information retrieval. WSD methods either rely on an external knowledge base (e.g., a thesaurus), on a corpus in combination with a machine learning algorithm, or on a combination (cf. Manning & Schütze, 1999; Sanderson, 2000). A problem with most of these methods is that they do not scale well. Several authors have therefore investigated whether and to what extent lexical ambiguity deteriorates retrieval performance or to what extent WSD can improve IR performance.

An extensive study was conducted by Krovetz & Croft (1992), using the CACM and TIME test collections. They manually disambiguated word senses of the query terms and counted word-sense mismatches in the top ten retrieved documents for each query. It was found that sense mismatches occurred more often in documents that were judged non-relevant than in relevant documents. Removing those irrelevant documents manually resulted in a small improvement in P@10. Surprisingly, only very few sense mismatches (ca. 10%) occurred in those top ten documents. Krovetz & Croft did a further analysis and found that two factors contributed to this effect: (i) many query terms are used in their most frequent sense, in the domain specific collection this is probably also the prevailing sense; (ii) top ranked documents will contain many query terms (due to the coordination effect of the retrieval model), which induces a sense match by context. The fact that there were very few sense mismatches in the top ten means that the potential effect of WSD is limited.

Another explorative study into the potential of WSD for IR was carried out by Sanderson (1994). He reused the pseudo-word simulation method, originally proposed by Gale et al. (1992), to introduce artificial ambiguity in a document by the concatenation of (random) index terms e.g., *banana@kalashnikov*. The advantage of this method is that disambiguation is trivial (since the original documents provide the ground truth) and the amount of ambiguity and accuracy of disambiguation can be controlled. The experiment showed that WSD accuracy should be at least 90% in order to improve retrieval performance. The positive effect was only noticeable for short queries. A follow-up study confirmed that the artificial ambiguity is indeed a good model for real-world ambiguity and that the two factors already put forward in Krovetz & Croft (1992) (frequent use of most frequent sense and collocation effect) are indeed the main reason for the limited potential of WSD for IR.

Mihalcea & Moldovan (2000) present an experiment in which a small IR test collection (Cranfield) has been processed by a WSD module, which assigns senses from Wordnet. About 55% of the nouns and verbs were disambiguated with an accuracy over 92%. A retrieval run based on a combination of a word and sense representation yielded a relative improvement of 4% in precision and 16% in recall. The authors make clear that their WSD algorithm does not scale well to very large collections.

A different approach is to take the document collection itself as a resource for the definition of word senses. Schütze & Pedersen (1995) argue that word senses as defined in dictionaries are often too fine grained for IR purposes. They present an experiment where word senses are derived from a clustering process based on the context of ambiguous words. Experiments with the WSJ part of the TREC-1 collection result in a 7-11% improvement w.r.t. the baseline.

We can conclude that the potential of WSD for IR is relatively low, although there might be some possible gain for shorter queries. Secondly, WSD techniques are computationally rather expensive. Both conclusions explain why WSD has not become a standard module in state-of-the-art IR systems.

3.4. Stop lists

Stop lists are a standard component of most IR systems, but have not received a lot of attention in IR literature. The only publication that we are aware of is (Fox, 1990). Still, the composition of a stop list can have significant impact of retrieval performance. A stop list is employed as a filter during indexing. Candidate index terms that are listed in the stop list are ignored during indexing. Since the creation of a stop list is not a trivial activity and a stop list had to be created for experiments with a Dutch test collection, we discuss the different approaches for creating stop lists.

Stop lists are lists consisting of "insignificant" words, words that do not contribute to the meaning of a document or query. This definition can be criticised, because there are hardly words without meaning. But if this question is considered from the bag-of-words model perspective, things change. Words which are a member of closed classes do not contribute significantly to the "semantic profile" of a document since they do not discriminate well. There are several reasons to use stop lists. The first reason is efficiency. Stopped indexes are much more compact because a large share of the tokens in a text is produced by a small fraction of types (Zipf's law, cf. Section 2.2.1). A reduced index size speeds up both indexing and retrieval time, because less postings have to be processed. A second reason is to remove terms from the query and documents to avoid matches based on query terms that do not discriminate well or belong to query phrasing e.g., "Relevant documents should discuss". . However, one could argue that this function (avoiding matches based on non-content terms) should be taken care of by the IR model. Indeed, this is the exact function of the idf component which is present in some form in all statistical IR models. One could also say that by removing frequent terms that do not carry meaning, the document ranking is hardly affected (Hiemstra, 2001), thus stop word removal is harmless from that point of view. On the other hand, there are a lot of infrequent function words (e.g., "daarentegen") which would seriously hurt retrieval precision if they would be used in the query and would not be stopped. Hiemstra's language-model-based IR approach obviates the application of stop lists by introducing query term specific importance weights, which can be trained through relevance feedback. We think that stop words can only be properly incorporated in IR models when queries are analysed at a higher level than the common practice to treat all query terms equal by creating a bag-of-words. Most current IR engines apply stop word removal in a very modest way, because stop words are often important for "exact phrase searches", like *"To be or not to be"*.

There are three different approaches for the construction of stop lists:

Functional: All members of closed classes are removed. Note that homographs are quite frequent among these words. A conservative approach, which is restricted to non-homographs is to be preferred in order to prevent a loss in recall. Another option is to use a POS-tagger to help to disambiguate those cases.

Corpus specific/Frequency-based: All terms with a document frequency higher than a certain threshold are removed.

Query specific: Query specific phrases e.g., "Find documents that discuss...." are removed from the query.

Usually the stop function is applied before stemming, thus a stop list must include inflected forms of e.g., auxiliary verbs. However, when a POS-tagger is part of the indexing process, the stop function has to be applied after POS-tagging/lemmatisation. In this case, the stop list consists of lemmas, possibly with part-of-speech information. The latter approach has the advantage that a conservative approach in the construction of stop lists is not necessary.

During the UPLIFT project, we used several stop lists. Some of them are described in table 3.4, complemented with the mean average precision as measured on the UPLIFT test collection using the Okapi BM25 weighting scheme. Both runs with a stop list perform significantly better than the run without a stop list (at the 0.01 level). Apparently the stop list based on closed classes and corpus frequencies is the most effective (also significant at the 0.01 level). We analyzed a few topics with marked differences, and the largest differences were due to the removal of stop phrases by the "1326" stop list. It is interesting to note that applying a better stop list can result in improvements on a similar scale as applying better term weighting algorithms (cf. section 6.1.2).

stop list size	m.a.p.	run description
0	0.296	baseline: no stop list
1326	0.322 (+9%)	The original stop list, based on a combination of linguistic and frequency criteria (both documents and queries).
1705	0.307 (+4%)	Based on closed classes in the CELEX machine readable dictionary. This stop list is used for the experiments in chapter 6.

Table 3.4. Mean average precision as measured on the UPLIFT collection, using several stop lists

3.5. Conclusions

In this chapter we have discussed several techniques to enrich poor data that are often seen as external to IR models. All techniques have the goal to maximize the match between query and relevant documents (recall enhancement) or to minimize the match between the query and irrelevant documents (precision enhancement). A very active area

of research is query expansion. Especially short queries, like the ones submitted to Web search engines can benefit from these techniques.

We have discussed methods that exploit term co-occurrence patterns. Co-occurrence patterns are indicators for semantic relationships between terms. These relations can be discovered off-line (e.g., similarity thesauri), resulting in global associations. It is even more effective to find related terms on-line, using a form of local feedback. The advantage of this approach is that only topic related associations are found. These associations are thus more specific than the global associations, because they are based on co-occurrence with more than one of the query terms. As is true for most data-oriented approaches, here too it holds that the more data is available, the more effective the method is. Thus, query expansion based on a secondary large collection improves performance even more. Expansion with query term synonyms extracted from hand-crafted thesauri can also improve performance, although this method is not so successful as corpus-based query expansion. A second recall enhancement tool is approximate string matching. This technique can either be used to overcome term mismatches due to spelling variation, typos and OCR errors but can also be used as a method for robust matching of morphological variants.

Techniques from the field of language technology have been applied in an IR context with varying success. Not all NLP techniques are ready for application on a large scale document collection. The purely statistical methods provide a high baseline, which can only be improved when linguistic analysis is highly accurate. The most successful application of linguistic techniques for IR is morphological analysis. Significant IR performance improvements have been achieved by the reduction of morphological variation (stemming, compound splitting). Effects are most pronounced for languages with a rich morphology.

All successful techniques based on query expansion have been designed to produce a balanced query, which does not distort the original weighting of the query terms. However, most of these weighting procedures are heuristic in nature and contain parameters that have to be tuned on a training collection. In our opinion, it would be better to integrate these techniques into a single framework. We will give some examples of an integrated approach based on a language modeling framework in the following chapters.

CHAPTER 4

Evaluation methodology for IR experiments

Because experimental validation of several retrieval models is an important element of this thesis, a proper evaluation methodology is essential. Evaluation is based on testing and comparing IR systems in which the different research hypotheses have been operationalised. The systems are tested on *performance measures* like precision or recall. In physical sciences and especially behavioural sciences, it is common practice to repeat measurements several times in order to improve the accuracy and reliability of the measurement. A series of measurements makes it possible to get some idea about the natural variation in the data and to determine the value of the desired measure with greater confidence. In IR, the situation is different. IR systems are completely deterministic. But the performance of an IR system for different queries can be quite different. To get a robust idea about the average performance of a system, the performance is measured over a set of queries in order to compute an average performance. Usually, the variation in retrieval performance across different queries is much larger than the variation of the averaged performance measure across systems (different hypotheses) because some queries are much harder than others for all systems. This calls for hypothesis testing techniques, which are able to detect consistent and significant performance differences between systems despite the noise introduced by query variation. We investigated whether standard statistical validation techniques that are common in experimental data with natural variation can also be applied for IR data. We have evaluated the core assumptions for several of these tests on experimental data based on the UPLIFT test collection. We also report about some quality assurance motivations concerning the development of the UPLIFT test collection.

Evaluation of IR systems has matured thanks to rigorous benchmarking tests like TREC and constructive criticism from the statistics community. The need for accurate statistical analysis of results has often been acknowledged by researchers, but the validity of most standard tests has been questioned (e.g., Rijsbergen (1979)). As a methodological justification, we will present a thorough overview of the evaluation process and present our choices concerning performance measures, test collections and experimental design with a special focus on the validity of statistical inference.

4.1. Evaluation Types

There are two main approaches to evaluation: (i) *glass box* evaluation, i.e. the systematic assessment of every component of a system and (ii) *black box* evaluation, testing the system as a whole (Group, 1996). Regarding the evaluation of a complete system, it is

not obvious how one could measure the vague notion of quality. There are again two main approaches: system oriented and user oriented evaluation.

The system oriented evaluation (of a complete IR system) has been the mainstream in evaluation since automated indexing and searching systems were developed in the 1960's. One of the major goals was to check whether these automatic systems performed as well as manual procedures (Sparck Jones & Willett, 1997c) . But there were also evaluations which compared the relative performance of indexing languages (the Cranfield and MEDLARS studies (Cleverdon, 1967; Lancaster, 1969)), or evaluations which compared different automatic indexing schemes (the SMART project).

The system-oriented evaluation has the advantage that experimental conditions can be highly controlled, using batch-mode experiments. There are however limitations to such an evaluation. A real-life information retrieval task comprises the full process of query formulation, query re-formulation and document selection. Current IR systems are equipped with graphical user-interfaces and offer many options for refining the query or restricting the result list. In order to measure the effectiveness of these interfaces, user oriented evaluations are required. Research has shown that improved user-interfaces can have a significant boosting effect on retrieval performance (Dumais, 1994). Unfortunately designing user centred evaluations has proven to be quite difficult. The annual TREC evaluation conference includes a program for the evaluation of interactive systems, but did not yet (TREC7-8-9) yield conclusive results. A key problem with interactive tests is that one cannot compare a test condition versus a control condition by asking the user to evaluate a system twice based on the same query. That means that an interactive experiment would require many more subjects and queries in order to average out inter-subject and inter-query variation and allow tests with a significance level comparable to an experiment without interaction. Then there is the problem of controlling variables. In an experiment one generally wants to measure the effect of a controlled variable on one or more dependent variables. But in an experimental context there are always other factors which can influence the dependent variables, and which are hard to control or are simply unknown. Examples of these are: computer skills, age, education, order effects etc. It is hard to come up with a representative group of subjects from which generalisations can be drawn.

Most evaluation experiments have the goal to allow statistical inference of this form: given the conditions of the experiment we can conclude that..., given the sampling methodology it's fair to assume that... For this thesis we have chosen to restrict ourselves to controlled experiments following the established tradition of automatic, batch retrieval runs. The reasons for this (commonly made) choice are: (i) results will have a greater comparability to other studies, (ii) the complexities of experimental design, which are a necessary component in user studies are avoided. We will review the tradition of batch experiments in section 4.2. In section 4.3 we introduce the performance measures that will be the leading dependent variables in our experiments. Section 4.4 discusses statistical methods for significance tests. Section 4.5 discusses the importance of pool quality for IR experiments and presents a quality assessment of the pool of the UPLIFT test collection for Dutch text. Section 4.6 summarises the evaluation methodology that we will apply in part II. For a more comprehensive overview of IR evaluation techniques

we refer to Sparck Jones (1981), Tague-Sutcliffe (1995), Salton & McGill (1983), the special issues on IR evaluation of Information Processing & Management(1992) and Journal of the American Society for Information Science(1996) and the section on evaluation in Sparck Jones & Willett (1997c).

4.2. System oriented evaluation

The technique of batch oriented retrieval evaluation and its associated performance measures has been developed in a number of long term research projects: Cranfield, MEDLARS, SMART, STAIRS and TREC. The main idea is to measure the performance of a retrieval system by running a set of queries on a collection of documents, indexed by the system, and recording the results. Now for each query, we can calculate the precision and recall of the recorded result set. As defined in chapter 1, precision is the fraction of relevant documents in the result set and recall is the fraction of the total amount of relevant documents in the collection, which has been retrieved. A more precise definition of these and related measures will be presented in section 4.3.

4.2.1. From Cranfield to TREC. The Cranfield project carried out by Cleverdon is often regarded as the role model for TREC (Cleverdon, 1967). Cleverdon created a "laboratory environment" for testing different indexing language devices (e.g. a device to promote recall or a device to improve precision) in isolation. He advocated *contrastive experiments*, where a single device was tested against a baseline, instead of comparing amalgams of different devices. The Cranfield *test collection* consists of abstracts of 1400 research papers in the field of aerodynamics. These papers were indexed manually using different indexing languages. Subsequently, 221 queries were produced by the original authors and all 1400 documents were judged on relevance for each of the queries, so complete *relevance judgements* are available for this collection. Cleverdon varied the coordination level of his queries by varying the scope of conjunctions in the query. E.g. a specific query is: '"A and B and C" , a less specific query is: "(A and B) or (B and C) or (A and C)", a loose query is: "A or B or C". This allowed him to make precision-recall plots. Because boolean systems do not produce a ranked result list, Cleverdon used this trick to produce a ranked result list, which is required to create the familiar precision-recall plots. Cleverdon used his test collection to compare different indexing languages by means of precision-recall plots. Although created in 1967, this test collection is still used by researchers today. The re-usability of test collections has proven to be a key issue for the development of IR technology and has become one of the safeguards of quality assurance in IR evaluation.

The MEDLARS study was one of the first evaluations of a fully operational system for searching in medical publications (Lancaster, 1969). Its setting was thus much more realistic than the laboratory setting of Cranfield. The MEDLARS test collection consisted of 800.000 citations (short abstracts) of articles in the medical domain. Indexing of the articles was done manually by using MeSH, a controlled index. Retrieval was fully automated. The scale of the collection forced the design of some new procedures in order to form a pool of relevant documents. Each test user (a professional in the medical domain) compared the results of his query with the relevant literature he was already aware of. Also librarians and authors in the field of the search request were consulted to create

a pool of good quality. The study showed that the MEDLARS system has an average of 50.4% precision and 57.7% of recall. The MEDLARS study is especially influential because of its extensive failure analysis. Lancaster investigated the relative influence of different system components on precision and recall failures. The factors studied were: indexing, searching, indexing language, user-system interaction. Formulation of a complete and precise search request proved to be of seminal importance.

The SMART project is probably the longest duration information retrieval study until today. Salton started research on information retrieval at Harvard in 1961. He wanted to develop a framework for systematic comparison of indexing and retrieval techniques. The framework was implemented by a series of algorithms and became known as the SMART system. Over the period of 1961 until Salton's death in 1996 the SMART group did experiments with every aspect of IR systems: term weighting, query expansion, relevance feedback, clustering etc. All these experiments were based on the SMART system (which was re-engineered for every new generation of computers). The SMART project resulted in the effective and intuitive vector space retrieval model. For a more technical discussion of the SMART project, cf. section 2.5.1. For a historical overview of the SMART project, cf. (Lesk et al., 1997).

Another influential study of an operational system is the STAIRS study (Blair & Maron, 1985; Blair, 1996). STAIRS was a commercially available IR system marketed by IBM. The study is famous for its finding that IR systems perform very poorly on recall. In contrast with MEDLARS and Cranfield, the STAIRS study is an evaluation of a full text IR system. In the study, the IR system was used as a litigation support system. The database consisted of 40.000 documents related to a lawsuit[1]. In such a situation, high recall is extremely important. The searchers had the predefined goal to locate at least 75% of all relevant documents. In reality they only found 20%, whereas the precision of their searches was 79%. The peformance of the STAIRS system is actually quite good compared to current TREC standards: a 79%, 20% P-R score would lie above the P-R plot of current state of the art systems in TREC. An important difference with TREC-style experiments is the definition of relevance used in STAIRS. Judges could assign four levels of relevance: "vital", "satisfactory", "marginally relevant" and "not relevant". In the precision and recall computation, marginally relevant documents were assumed to be relevant. This is defendable, because the lawyers stipulated that they need at least 75% of the relevant documents to prepare a case. In recent TREC evaluations, the definition of what constitutes a relevant document is much more restricted, which makes a direct comparison between the two studies difficult. The validity of the low recall levels could be explained by the fact that search requests operated on the wrong level of precision/recall trade-off. Sormunen (2000) concluded that the searchers probably were formulating high precision instead of high recall search requests. The original STAIRS paper describes that the search intermediaries were encouraged to continue the search process until they were convinced that they had enough information to defend the particular aspect of the lawsuit reflected in the query.

[1]The lawsuit concerned the construction of the BART railway in the San Francisco Bay area, cf. Blair (1996) for more details.

> In the test, each query required a number of revisions, and the lawyers were not generally satisfied until many retrieved sets of documents had been generated and evaluated (Blair & Maron, 1985).

This procedure seems to invalidate Sormunen's conclusion, but there is also some positive evidence:

> Another information request resulted in the identification of 3 key terms ... The 3 original key terms could not have been used individually as they would have retrieved 420 documents, or approximately 4000 pages of hard copy, an unreasonably large set, most of which contained irrelevant information (Blair & Maron, 1985)

It looks as if the search intermediaries adjusted their queries in order to avoid having to print out lots of irrelevant documents (which had to be evaluated by the lawyers). Still the STAIRS study has shown the limitations of the retrieval performance of automatic indexing systems for a large document collection. STAIRS adjusted the general opinion about recall levels of IR systems to a more realistic modest level. Blair & Maron mention the inability of users to foresee the exact words and phrases used in relevant documents and only in those documents[2]. The STAIRS study also necessitated the development of new strategies for recall measurement, as full relevance judgements were clearly not feasible anymore. Blair & Maron chose the following procedure: they constructed collection fragments they thought would be rich in relevant documents. The judges had to assess the relevance of samples of these fragments. Thus for each query, recall was estimated on a query specific document collection fragment. In a later analysis of the STAIRS experiment, Blair states that the study probably gives an upper bound of obtainable recall, since the conditions were much more favourable than under normal operational circumstances Blair (1996).

The ongoing TREC program has been inspired by the Cranfield and SMART studies. TREC started in 1992, with two main tasks: ad hoc searching and "routing" (a filtering task, which we will not discuss here). Since then, a lot of new tasks have been tested in several *tracks.* The main characteristics of TREC are that the collection sizes are much more realistic, and that the evaluation is open to any research group. Participation in TREC has increased steadily over the years. A significant number of groups has participated every year, which ensures stability and comparability over the years. TREC uses a board of assessors from the National Institute of Science and Technology (NIST) to perform relevance assessments. The STAIRS study was one of the first studies that had to develop a new procedure to measure recall, because the collection size made complete relevance judgements too costly. TREC also bases its recall measurements on judging just a subset of the documents (the pool) but uses a different method to construct it. The pool is created from a sample of different runs (as different as possible). For each query, the lists of retrieved documents of each run are combined (by merging and removing duplicates) resulting in a list of unique documents. Subsequently assessors judge for each document in these lists (there is a list for each query), whether they are relevant to the corresponding query. The influence of TREC on IR research is large, the quality of the

[2]The STAIRS query language includes Boolean operators, thus theoretically supports the composition of a perfect query, if the users have complete knowledge of the contents of the document collection.

test collections is good, since a lot of different participating systems contribute to the pool (cf. section 4.3.1) and because of the continuity of the program. TREC has produced a wealth of test collections which can be used for a variety of controlled experiments.

A great virtue of a controlled experiment is the fact that it can be replicated. Before TREC, a lot of relatively small test suites existed, which made it hard to compare approaches between different groups. This situation prevented real progress in the field. TREC had the goal to build a number of large test suites for IR, essentially to (i) perform experiments under controlled conditions, (ii) build test suites enabling the replication of experiments.

Looking at the results of groups that did participate in TREC from the start, one can observe quite substantial improvement. In a study where the TREC-1–7 versions of the SMART system were tested on the test collections of TREC-1–7, improvements between 50% and 124% are reported (Buckley et al., 1999).

One would hope that performance increases of this size are noticeable as well in an interactive user session, but seen from a user's perspective, quality is dependent on many more aspects. A recent study, has shown that improvements in retrieval effectiveness as measured in a batch evaluation cannot always be detected in an interactive situation (Hersh et al., 2000). He compared the Okapi system with a plain *tf.idf* system. It is well known that Okapi is a much more effective system than *tf.idf* in batch TREC evaluations. Hersh compared both systems in an interactive setting. 24 subjects performed an *instance recall* [3] task on six different topics. A statistical significance test (analysis of variance) showed that the Okapi system did not improve the instance recall of the subjects. Although the batch and interactive retrieval experiments are evaluated by different measures and the topic collection of the interactive experiment is rather small (6 topics), it seems not self-evident that differences in retrieval performance carry over to interactive systems. More extensive research is needed to answer this question. Still, batch evaluations with standardized test collections are of great value for IR research.

4.2.2. Evaluation procedure. In the common batch-oriented evaluation practice the following steps can be distinguished:

(1) *Build a test collection* A test collection consists of a set of documents, a set of topics (a description of the search request) and *relevance judgements*. Ideally each query–document combination is tested for relevance. In practice, usually only a part of the document collection has been judged for each query. Appendix C.2 describes how such a collection fragment is selected.
(2) *Test systems on the test collection* Index the document collection, construct queries from the topics, retrieve a *relevance ranked* list of documents for each query.
(3) *Compute performance measures* The classical measures are *precision* and *recall*, but there exist numerous other measures. Currently the most commonly used measure is *mean average precision* (map or avp).
(4) *Assess the significance of the data with statistical tests* The global performance measures are essentially averaged over the query set. Because the variability of

[3] An example of an instance recall task is the query "Which countries import Cuban sugar?". Subjects have to find documents which mention countries that import Cuban sugar.

queries is huge, the variance of the calculated measures is quite high. A proper statistical analysis is required to assess whether the differences measured between systems are statistically significant to a certain confidence level.

4.2.3. Relevance assessments. In TREC style IR evaluation, two important assumptions are made, which probably do not hold in most real-life settings:

(1) Relevance is an absolute notion: a document is either relevant or not relevant.
(2) The relevance of a document is independent of other documents.

These assumptions simplify the measurement of retrieval performance. Several researchers are experimenting with a more refined relevance scale. A 3-level scale was used in NTCIR 1999 (NII-NACSIS Test Collection for IR Systems), the Japanese edition of TREC (Kando & Nozue, 1999) and the WEB track of TREC-9. A 4 level scale was used in NTCIR 2000 and in a study at the university of Tampere (Järvelin & Kekäläinen, 2000). In STAIRS, the different levels were projected onto a binary scheme for final evaluation. NTCIR 1999 did separate evaluations for different relevance levels, but it was found that the results were highly correlated. A new evaluation measure which takes the relevance levels into account has been proposed by Järvelin & Kekäläinen (2000). This measure, the so-called discounted cumulative gain (DCG), models the utility of a ranked retrieval list for a user. The DCG of a run is defined as the sum of the relevance of the retrieved documents (the relevance score ranges from 0 for not relevant to 3 for highly relevant). The relevance score of a document at rank N is discounted by a function of its rank number, reflecting the fact that documents which are retrieved further down on the list, are probably less valuable for the user (due to limited search time, effort and redundancy in information). The discount function is a log of the rank number. The main motivation for this new measure, is that it can discriminate systems that are able to rank highly relevant documents near the top of the ranked list. A disadvantage of the approach is that the average DCG is biased towards topics with a lot of relevant documents.

The assumption that the relevance of a document is independent of other documents is not realistic in most cases. In most elementary information-seeking tasks, like search on the web, searchers look for an answer to a particular question or for some good references. Suppose a user will start browsing through the retrieved documents starting with the most salient document; then the relevance of documents further down the list (thought in terms of utility) is dependent on the documents already read. The probability that a document will contain new information will decrease further down the list. This dependency is usually ignored by IR researchers. An exception is the interactive track of TREC, where subjects are asked to find and mark relevant documents which discuss different aspects of a search topic within 20 minutes. The search topics were especially constructed to target *a list* of answers, e.g. "Which treatments can ...". The performance was measured in terms of aspectual recall, which is defined as the proportion of (ground truth) aspects found (Over, 1997). In addition, the recent novelty track at TREC aims at reducing redundancy in a passage-retrieval task. Some Web search engines group near-duplicates, in order to improve user satisfaction.

An example of a study where the conditional relevance of documents is modelled is Carbonell & Goldstein (1998). Here, the final document score is composed by the initial

score subtracted by the maximum similarity of the document with the documents which have already been presented to the user.

There have also been concerns about the subjectivity of the assessment procedure. Humans often have different opinions about relevance. This could have a negative impact on the robustness of the TREC evaluations. However, there have been several studies which addressed this problem and found that the influence on the relative ranking of systems is negligible (e.g. Burgin, 1992). A recent study concerning the TREC collection tested a lot of different factors (Voorhees, 1998):

- judgements by authors vs. non-authors
- judgements by a single judge vs. group judgements
- judgements in the same environment vs. judgements in very different environments[4]

These factors influenced the absolute values of the performance measures, but the relative ordering of systems remained stable, even variants of the same underlying IR system.

Another concern was that the TREC collections would be biased towards judged runs, and that the collection would not be usable for new systems that did not contribute to the pool. A recent study showed that indeed the TREC pooling procedure is adequate. The TREC judgement pool is shown to produce reliable measurement results, also for new systems (Zobel, 1998). Zobel argues that NIST resources for assessment could be more efficiently used by judging more documents for topics with a lot of relevant documents. The number of relevant documents can be reliably estimated during the assessment procedure, by identifying the good systems at an early stage (Keenan et al., 2001). So far, the Zobel method has not been applied in an IR evaluation, probably due to its higher complexity.

4.3. Performance Measures

The classical performance measures for IR experiments are recall and precision.They were originally introduced in Perry et al. (1956) for IR systems that retrieve an (unordered) set of relevant documents. There are several other measures such as Swets' E-measure (Swets, 1969) or the average search length. In the following subsections we will describe the procedures for measuring recall and precision and derived measures for ranked retrieval systems in a situation where it is impractical to assess[5] all documents in a collection. For more background on other performance measures we refer the reader to (Rijsbergen, 1979) and (Salton & McGill, 1983).

4.3.1. Measuring recall. The computation of recall is a well-known problem in IR evaluation, because it involves the manual assessment or estimation of the total number of relevant documents in the database for each query. Assessment of each document is too

[4]Judgements for the same test collection made by two different organisations: NIST and the University of Waterloo.

[5]The result of the process of deciding whether a document is relevant for a certain topic/query are the so-called *relevance judgements* or *relevance assessments*. The people involved in this process are usually called *assessors*.

costly for large document collections and estimation by assessing a sample with sufficient reliability would still require large samples (Tague, 1981). For the UPLIFT collection, a test collection consisting of Dutch newspaper articles which is described in appendix C, we decided to use the 'pooling method' which is applied in TREC. This method computes *relative* recall values instead of *absolute* recall. It is assumed that if we have a 'pool' of diverse IR systems, the probability that a relevant document will be retrieved by one of the systems is high. So a merged list of document rankings (cf. is assumed to contain most relevant documents. The pool assumption is actually a bit more precise: we assume that most relevant documents are contained in a pool consisting of the merged top D documents of several different high quality IR systems. Here D is the *pool depth* i.e. the number of documents taken from the top of a retrieval run. For our experiments the standard TREC pool depth of 100 documents has been applied.

Since the result sets of ranked retrieval sets are ordered, precision and recall can be computed at several document cut-off levels, by taking the top N documents from a result list. Measuring at a certain document cut-off level shows the trade-off between recall and precision: if the document cut-off level is increased, recall increases as well, but precision decreases. It is exactly this interaction which is depicted in the so-called recall precision graphs, which will be discussed in the next section. However, these graphs are based on precision at fixed recall levels instead of document cut-off levels, to enable averaging over queries. Averaging of precision at fixed document cut-off levels is problematic because the number of relevant documents per query usually varies a lot. Measuring precision@100 (precision at a document cut-off lever of 100) for a query with 15 relevant documents has a maximum value of 0.15, unlike a query with say 500 relevant documents. Averaging precision over precision over fixed recall levels overcomes this problem.

4.3.2. Precision vs. recall curve. A concise and perspicuous way to present the performance of a retrieval run is a graph where precision is plotted as a function of recall (PR curve).

The basis for the computation of data for a PR curve is formed by the relevance judgements and the ranked document lists as produced by the IR system for each query. It is easy to compute recall and precision for each rank in the list. It is not so easy however to compute the average precision as a function of recall across topics, since each topic has a different number of relevant documents. One way to average precision values over a set of queries is to compute interpolated precision values at fixed points of recall. A standardly adopted interpolation algorithm is the one implemented in `trec_eval`, which is distributed as part of the SMART IR evaluation suite. At each fixed recall point the interpolated precision is defined as the maximum of the precision-at-fixed-recall points greater than or equal to the recall value in question.

$$\text{pr}(i) = \max(\text{pr}(j)) \quad \text{where} \quad j \geq i \tag{51}$$

The interpolated data can be used to compute precision at eleven standard points: 0, 0.1, 0.2, ... 0.9, 1.0. Salton & McGill (1983) give a detailed account of this procedure. Figure 4.1 gives an example of such a PR curve. Interpolation thus forms the basis for PR curves. One can also average the precision over the 11 standard points of recall: average 11-point interpolated precision. This precision measure is not recommended, because

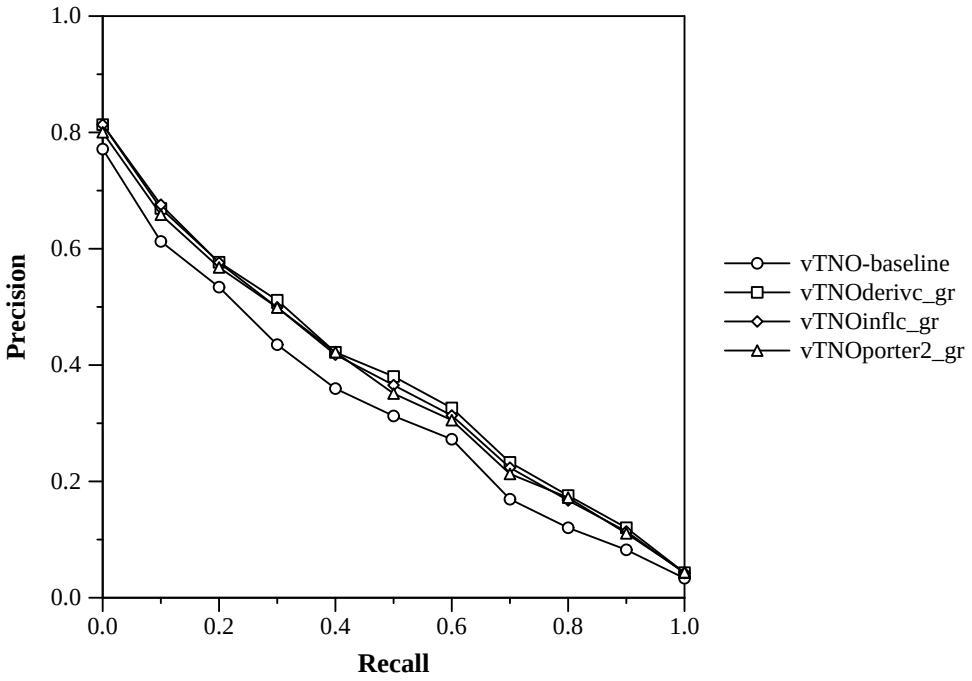

Figure 4.1. Example of a PR curve.

it is strictly based on interpolated data. A method of averaging which is more faithful to the actual data is mean average precision, also referred to as average *un*interpolated precision (cf. section 4.3.4).

4.3.3. Ties. A problem arises when several documents have an identical retrieval status value (RSV). In this case a tie-breaking procedure is needed since the evaluation procedures can only handle completely ranked document lists. A common technique is to use the document-id as a secondary sort criterion. The tie-problem can affect the reliability of measurements in a substantial way, especially if the ties are long. In earlier work (e.g. (Kraaij & Pohlmann, 1996b)) we experimented with stemming based on query expansion. These experiments yielded result lists with extremely long ties. We introduce the measure *resolving power* in order to quantify the sensitivity of a particular IR system to the tie-problem. The resolving power of a system is defined as the average number of different scores (RSV's) per rank, averaged over all queries of a particular run. We computed the resolving power of several variants of the main IR systems, used for experiments with the UPLIFT test collection (cf. chapter 6 and appendix C). Table 4.1 lists the resolving power for 4 different classes of runs. The classes comprise runs with two different retrieval systems: TRU and TNO (cf. section 6.1.2 for a description of these systems), each in a basic setting and a setting with query expansion.

When we look at the order of magnitude of resolving power for these classes, we find huge differences: The low resolving power of the TRU runs is caused by ties with

System	range of resolving power (%)
TRU-standard	3.1-3.6
TRU+expansion	1.6-3.1
TNO-standard	79.0-92.3
TNO+expansion	75.5-98.3

Table 4.1. Resolving power of different system classes

an average length of 30-60 (depending on the run). This means that in the evaluation procedure, in the worst case, we have an uncertainty of approximately 60 ranks, which directly affects the reliability of precision and recall measurements. The reason for the long ties in the TRU engine were twofold: (i) a too economic representation of term weights (4 bits, allowing for just 15 distinct levels of a term weight) and (ii) an application of length normalisation on queries, which had a bad side effect when used in combination with query expansion. We will discuss the different term weighting algorithms in more detail in section 6.1.2. Fortunately, the TNO engine has a much higher resolving power, ranks contain on average 1.1 documents.

In previous publications (e.g., Kraaij & Pohlmann, 1995, 1996b) based on the TRU engine, ties were handled in the following way: if a tie contained relevant documents, these were moved in the middle of the tied group. We have also experimented with more sophisticated approaches (Raghavan & Jung, 1989) with ties explicitly modeled in the evaluation metric, but in this thesis we strictly use `trec_eval` in order to conform to standards and because the TNO search engine implementation produces only short ties. `trec_eval` breaks ties by a secondary lexicographic ordering on document id.

In chapter 6 we report some TRU runs. The reader should bear in mind that it is hard to discriminate between system versions, because of the low resolving power of the TRU engine.

4.3.4. Mean Average Precision. Whereas precision-recall plots give a quite detailed impression of the quality of a system, it is often practical to have a single figure for the performance quality. One possibility is to average the 11 precision values which make up the precision-recall plot. This has the disadvantage that all these values are interpolated and thus less reliable, especially when a query yields only a small amount of relevant documents. The average uninterpolated precision does not suffer from this problem. The terminology used for this term is not fully standard; most researchers shorten it to 'average precision' (AvP), but recently the term 'mean average precision' (MAP) has become popular, which reflects the fact that the computation is a result of two averaging steps.

The average precision for a certain query and a certain system version can be computed by identifying the rank number n of each relevant document in a retrieval run. The corresponding precision is defined as the number of relevant documents found in the ranks equal or higher than the respective rank r divided by n. Relevant documents which are not retrieved receive a precision of zero. The average precision for a certain query is defined as the average value of the precision over all relevant documents. The mean average precision can be calculated by averaging the average precision over all

queries (macro-average).

(52)
$$\text{MAP} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N_j}\sum_{i=1}^{N_j}\text{pr}(d_{ij}) \quad \text{where} \quad \text{pr}(d_{ij}) = \begin{cases} \frac{r_{n_i}}{n_i} & \text{if } d_{ij} \text{ retrieved and } n_i \leq C \\ 0 & \text{in other cases} \end{cases}$$

Here, n_i denotes the rank of the document d_{ij} which has been retrieved and is relevant for query j, r_{n_i} is the number of relevant documents found at ranks $1-i$, N_j is the total number of relevant documents of query j, M is the total number of queries and C is the cut-off rank (C is 1000 for TREC). The average precision for each query has the same weight in the calculation of the overall average precision. This procedure has the effect that the mean average precision is quite sensitive to topics with only a few relevant documents. For these "hard" queries, the relatively minor change of e.g. a relevant document from position 2 to position 6, can have a large consequence on the average precision of that query and indirectly on the mean average precision, although such a change is probably of no effect from a user's perspective. Several researchers have proposed to use a micro-averaging approach where precision is averaged in just one step over all relevant documents.

(53)
$$\text{micro average precision} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N_j}\text{pr}(d_{ij})}{\sum_{j=1}^{M}N_j}$$

This approach is less sensitive to "noise" caused by "hard" queries, because here each relevant document has the same weight in the overall average. The disadvantage of such an approach, however, is that the system performance is now dominated by "easy" queries with a lot of relevant documents, which is usually not desirable.

Mean average precision is easy to compute and has proven to yield reliable results in cross-measure evaluation experiments (Tague-Sutcliffe & Blustein, 1995). The measure has become the standard "single figure metric" in the IR community. The mean average precision has proven to be a suitable measure to make quick comparisons between a large number of system versions. Since our experiments have been constructed according to the TREC framework, we decided to use MAP as the basic performance measure. In addition, we selected measures aimed at measuring high-precision and recall for experiments that are designed to improve upon either of these. In particular we selected P@5-15 and R-recall. These measures will be discussed in the following sections.

4.3.5. P@5-15. Hull (1996) argues for a special measure which is tailored to measuring high precision, that is the part of retrieval performance which is probably the most visible to users. Hull used the average of the precision measured at 5, 10 and 15 documents (as computed by `trec_eval`). The averaging procedure produces a more stable measure than e.g. precision at 10 documents. We will denote this measure by P@5-15.

4.3.6. R-recall. Since recall measured at document cut-off levels of 200 or more seems only of importance for researchers and not for users, we experimented with recall at document cut-off levels of 25, 50, and 100. A disadvantage of this method is that "recall at 25" does not make much sense for queries with many or just a few relevant documents. The number of relevant documents for the queries in the UPLIFT test collection varied from 3 to 187. This variety motivated Kraaij & Pohlmann to measure recall at R

documents, where R is the number of relevant documents for a particular query(Kraaij & Pohlmann, 1996b). They call this measure R-recall. This measure is more intuitive since it normalises for query variance. An ideal system has an R-recall of 1 and R-recall is by definition equal to R-precision, which measure was independently introduced by Chris Buckley (Cornell University) for TREC2. R-recall thus provides a singular performance measure in which both recall and precision are expressed.

4.3.7. Discussion. The selection of good performance measures is still an area of ongoing research. There have been studies which have shown that there is a large correlation between the measures which are computed by `trec_eval`[6] (Tague-Sutcliffe, 1995; Voorhees & Harman, 1999a). As a consequence, the presentation of many alternative measures is not very informative. In a recent study, Buckley & Voorhees (2000) investigated the robustness of the common performance measures, including mean average precision, R-precision, precision at N and Recall(1000). They used a special test collection consisting of 21 different query variants for each of a set of 50 topics (thus a total of 1050 queries), which were run by 9 different systems. The interesting idea here is that the number of relevant documents for each topic is fixed, but of course there is a variability in performance level between the different query versions. This variability is used to measure the consistency of several performance measures. If a system scores consistently better than another system on all different query-sets, the performance measure is more consistent. Despite the fact that the chosen decision criterion for significance is rather arbitrary (differences of more than 5% are "worth noting"), the experiments indicate that mean average precision and R-precision are the most consistent and discriminating measures.

4.3.8. Conclusions. We have motivated the choice for the performance measures which have been used in our experiments: precision-recall plots for the overview, mean average precision for overall performance, R-recall for recall and prec@5-15 for high precision. MAP and R-recall have proven to be robust and stable measures, with respect to the relative ordering of systems. We have shown that the problem of ties can be safely ignored for the experiments based on the TNO engine, because the average tie length is close to 1. To ensure compatibility with the TREC series of evaluation experiments, we have used `trec_eval`[7] for the computation of all these measures.

4.4. Statistical validation

Statistical analysis of IR evaluation data recently has been given more attention. Suppose we would like to know whether the average un-interpolated precision of a system with stemming method A is significantly better than the same system with stemming method B. If, after calculating means, we find just a small difference, intuitively we will not draw firm conclusions about the superiority of either one of the methods. After all, the difference in means could be caused by chance (i.e. the data points will show some natural variation which is not due to the controlled variable, in this case the stemming

[6] `trec_eval` computes interpolated Recall - Precision pairs, average uninterpolated precision, precision at fixed document cut-off levels and R-precision

[7] version 3

method) or by outliers (measurement or experiment errors, e.g. a certain topic causes an exception in the processing stage and renders an effectively empty query).

In IR experiments, means are calculated over a set of queries as a sample from the population of all possible queries. This sample of measurements (e.g. a sample of average precision values) usually exhibits a high variance due to differences between topics. Some queries are 'easy', some are 'hard', which is reflected in widely differing performance differences for a system across topics. The 'across topic difference' accounts for most of the afore mentioned natural variance. Statistical significance tests help to prove that differences between means of the observed statistic are really due to the controlled variable (in this case IR systems) and not to chance. Significance tests help to draw well-founded conclusions instead. These conclusions can very well be counter-intuitive: a very small difference in means can be significant (because the differences between pairs of observations are consistent) , and a very large difference between means could turn out to be not significant. It could be caused by a high variance of the observations or a single outlier query.

We will start this section with the discussion of the basic concepts and methods for statistical hypothesis testing. After this general introduction, we discuss the methods for significance tests that are commonly used in IR experiments and motivate the choices we made for our experiments.

4.4.1. Introduction to hypothesis testing. There are many statistical significance tests, each designed for a particular setting and with specific assumptions about the data. In general, tests that have stronger assumptions about the data (e.g., parametric tests) are more powerful. This means that a more powerful test can detect smaller significant differences or requires less data to draw the same conclusion. Powerful tests are attractive for IR experiments, since the construction of large test collections is costly. However, the applicability of parametric tests for IR experimental data is controversial.

Section 4.4.1.1 introduces the main concepts of statistical hypothesis testing. Sections 4.4.2and 4.4.3 discuss two typical situations: the comparison of two samples and the comparison of more than two samples. Both situations have their corresponding tests. We will discuss several of these significance tests and motivate the choices we have made for our own experimental validation.

A more comprehensive treatment of statistical tests can be found in e.g. Hayes (1981), Maxwell & Delaney (1990), Stuart & Ord (1987) or Snedecor & Cochran (1980).

4.4.1.1. *Definition of statistical hypothesis testing.* Significance tests are a particular form of hypothesis testing. The question is whether we, given the observed data, can safely assume that a certain hypothesis is true, or that we have to reject this hypothesis. Hypothesis tests have the following basic structure: there are two hypotheses H_0 and H_1. Usually H_0 states that there is no effect, and H_1 that there is an effect. For example hypothesis H_0

$$H_0 : \mu_A - \mu_B = 0$$

could represent the hypothesis that there is no significant difference between the mean average precision of system A (μ_A) and B (μ_B). Hypothesis H_1

$$H_1 : \mu_A - \mu_B \neq 0$$

could represent the hypothesis that there is a significant difference (for a bidirectional test),or H_1 could state

$$H_1 : \mu_A - \mu_B > 0$$

that the mean average precision of system A is larger than system B (a unidirectional test).

In hypothesis tests, two types of errors can occur:

type I: Accept hypothesis H_1 when hypothesis H_0 is true. Before the test is performed, an error threshold for type I error must be chosen, this error threshold is called α.

type II: Accept hypothesis H_0 is when hypothesis H_1 is true. A low type II error means that a test is sensitive or powerful. One usually refers to the power of a test, defined as $(1 - \beta)$, instead of the type II error itself.

An ideal test would have low values for both type I and type II errors, but as usual there is a trade-off. A lower α level will decrease the power of the test. The power of a test is also dependent on the a priori knowledge about the properties of the data. If for example, we have good reasons to assume that a data sample has a normal distribution, we can use tests which are much more sensitive because they exploit the characteristics of the normal distribution. Also, the use of more data points (more queries in the IR case) increase the power of a test. A low α is important because we want to make statements about significant differences with a certain accuracy. A high power is important because otherwise we might not be able to detect meaningful differences at all. Usually experimenters work with $\alpha = 0.05$. This means that if the test rejects H_0 we can conclude that there is a significant effect with an accuracy of 95%. Usually the power of a test is not known, because power analysis is complicated.

4.4.1.2. *Common test settings.* In an IR experiment we are usually interested in finding whether there is some association between the dependent variable being measured (e.g. mean average precision) and a controlled variable e.g. a particular type of stemming. Controlled variables are also referred to as *factors*.

We will start with an inventory of some common controlled test situations and associate these with relevant IR evaluation cases.

Single factor, 2 levels: In this case for example 2 stemming algorithms are compared. The standard solution for comparing two means is to apply a paired t-test. In IR research, the application of a paired t-test is criticised. Cf. section 4.4.2 for a more detailed discussion.

Single factor, multiple levels: In this case we want to compare multiple stemming algorithms. The classical solution is to apply linear models. Section 4.4.3 discusses this and other options.

Multiple factors, multiple levels: The experimenter wants to compare a variety of systems, each of which has a different level signature for 2 or more factors. This situation calls for more complicated experimental designs (factorial or nested). Although it is common practice in (IR) experiments to perform contrastive experiments where just one factor is tested, it could be desirable to do higher order experiments if factors are not completely independent. We will

refrain from a discussion of this type of experiment here, because interaction of effects has not been studied in the experiments reported in part II.

An important case which is not covered by any of the classes listed above is the TREC evaluation across systems of different sites. At TREC, multiple systems are compared without detailed knowledge of the characteristics of the individual systems. We discuss this case in some detail, because in some respects it is relevant for the evaluation of the experiments described in part II. In the TREC situation, a lot of systems are almost similar because they are based on one of the popular term weighting algorithms like *BM25* or *Lnu.ltu*. Therefore a lot of systems exhibit a strong correlation of results and only a few systems are quite different. The fact that the set of tested systems consists of one or two clusters and some odd systems, has some side-effects on the significance tests. Some statistical tests are inappropriate for the TREC setting because they assume independence. Yeh (2000) shows that tests based on sample variance which falsely assume independence produce unreliable results[8]. If systems are heavily correlated, this will erroneously reduce the error term, because its computation assumes independence, which will have the effect that the threshold to conclude for a significant effect is decreased.

The default design for the TREC evaluation across systems of different sites is to model the systems as different levels of the factor *system*, because we do not have exhaustive knowledge of the essential factors and levels that determine the effectiveness of the individual runs. But as we have pointed out, the reliability of statistical inferences is significantly reduced when the set of systems contains a lot of dependency. One idea to is to remove as much of the dependency as possible, e.g. by clustering runs and take only one run per cluster, an idea which has been explored by Kantor[9].

The experiments presented in part II do not have the scale of TREC, so there is much more control over the different factors and their levels. But the problem of dependency between systems is certainly applicable.

In the following subsections we will discuss possible options for the classes of tests which we want to apply to our research: the comparison of two means and the comparison of n means (after Hull, 1993) .

4.4.1.3. *Types of tests.* Significance tests cannot only be classified according to the various test settings but also according to the assumptions about the data distribution which are postulated a priori. In a study on the evaluation of TREC results, Hull (Hull et al., 1999) compared three classes of methods for hypothesis testing. All tests are based on the general idea of computing the probability that the observed data samples could be generated by the null hypothesis (no difference between systems). The three classes of tests to be distinguished are:

(1) **Parametric tests.** Parametric tests owe their name to the fact that they assume that the error distribution of the data can be approximated by a parameterised standard distribution, usually the normal distribution. The test statistic is assumed to be composed of a population mean, a treatment effect, a system effect, possibly effects due to interaction between factors and a residual error

[8]Though he makes a mistake by claiming that the error term is increased by a positive correlation, instead of a negative correlation.

[9] cf. http://scils.rutgers.edu/~kantor/dizhao/html/adhoc/trec4.html

component. Parametric tests like student T or analysis of variance are based on the comparison of the variation due to a certain factor with the variation in the residual error term.

(2) **Non-Parametric rank tests.** These types of tests do not assume a normal distribution. The original data is transformed into a rank order, reflecting the rank order of the specific query-system score in the total list of scores for the respective queries. Under the null hypothesis we would expect that the average rank for each system is about the same. In such an approach, the absolute value of the relative differences is ignored, which has the advantage that all queries have equal influence on the significance test. Non-parametric tests have the disadvantage that they do not work with the original data and hence cannot be used to make inference about absolute values.

(3) **Simulation tests based on re-sampling.** A third option which has recently been proposed for IR experiments is to use simulations to estimate the null distribution, thereby avoiding any a priori assumptions about the shape of the distribution. The idea is that the observations themselves are a representative sample of the population. We can simulate the null distribution by re-sampling from the observed query-system matrix. This works as follows: for each query (corresponding to a row in the matrix) we randomly res-ample (re-shuffle) the measurements and compute system means. The simulation is repeated for example 1000 times. The resulting data can be used as a basis for a test statistic, e.g. by counting the number of times that the measured system mean is higher than one of the 1000 simulated means. When we res-ample without replacement, this strategy amounts to generating permutations and is called a Monte-Carlo test. This method is applied in Hull et al. (1999). The variant where the re-sampling is done with replacement has been applied by Savoy (1997). This variant is also called the bootstrap method and is based on random re-sampling from the set of measurements. The advantage of the simulation methods is that no assumptions are made about the distributions of the original data. All re-sampling based tests are computationally expensive and not widely supported by standard statistical packages, which explains why they have not been extensively applied in earlier research[10].

Summarizing: care should be taken in applying standard parametric tests, because model assumptions are often not satisfied. Non-parametric tests make fewer assumptions about the data, which could make them weaker. Recently tests based on re-sampling have been proposed, which overcome some of the limitations of model based tests.

In sections 4.4.2 and 4.4.3 we will discuss some common tests in more detail. In 4.4.4 we will wrap up the discussion of significance testing and present the approach we will use for our experiments.

4.4.2. Comparing two classes of samples. The prototypical test situation consists of two classes of observations. The experimenter has created two situations which are

[10]More recently, a bootstrap package has become available for the (open source) statistical package R, bringing bootstrap analysis within reach of IR experimenters (Monz, 2003)

identical with the exception of one condition: the controlled variable. The experimenter wants to check whether the observed variable is in some way correlated with the controlled variable. As an example take an experiment to investigate whether frequent consumption of olive oil prevents cardiac diseases. The classical test for these type of tests is to take a sample of the test population and a sample of the control population and apply the t-test (cf. 4.4.2.1). This test assumes that the samples are independent. However, for IR data this is seldom the case. The samples for the test and control system are usually based on the same set of queries. And with reason: it is a form of experimental control. If a difference between the two data points is observed, this difference cannot be due to between query variance. This is an important advantage because the across-query variation is much larger than the between-system variation, making a between system comparison without matched pairs difficult.

Fortunately there are several significance tests that can deal with paired samples. We will assess the applicability of several tests to IR data in the following subsections. In these subsections the paired samples are represented by a bivariate random sample $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$, from which a sample of differences $D_1 \ldots, D_n$ is derived.

4.4.2.1. *t-test for matched pairs.* The standard parametric test for a comparison of two samples is the t-test. It makes the following assumptions:

(1) The populations each have a normal distribution.
(2) Each population has the *same* variance σ^2.
(3) Samples are independent.

If these assumptions hold, one can use the t-test statistic to test whether there is a significant difference between samples. Just like the normal distribution, the distribution of t is a standardized score (the score is related to the sample mean and normalised by the variance) and serves to define confidence intervals for a certain estimated mean. Given the sample mean and sample variance, a confidence interval can be defined around the population mean. The form of the t distribution is exactly known only when the basic assumptions hold. As pointed out in the previous section, in the prototypical IR experiment samples are not independent, but samples are dependent pairs. This poses a problem for the standard t-test because the degree of dependence is unknown. However, the fact that samples are paired can be exploited: we can reduce the bivariate sample to a sample of differences: $D_1, ..., D_n$ where $D_i = X_i - Y_i$. The paired t-test can then be modelled by a test for a *single* mean:

$$t = \frac{M_D - E(M_D)}{s_D / \sqrt{n}} \tag{54}$$

Here M_D is the mean of the sample D, s_D is the sample variance. (cf. Hull (1993) or Hayes (1981) for more detail)

The (paired) t-test is not often used in IR , because raw IR measures like precision@10 or recall@1000 are far from continuous and normal. However, as argued in Hull (1993), averaged measures like mean average precision behave much more like a continuous variable. The t-test is also quite robust against violations of the normality assumption, as long as there are no big outliers and the distribution is symmetric. This can be checked with quantile plots.

As a case study we will compare two conflation techniques using a matched-pair t-test. The conflation techniques are both dictionary based. Technique 'vc1' removes derivational and inflectional suffixes; technique 'vc1f' removes only inflectional suffixes. Cf. chapter 6 for a further discussion of conflation techniques. We made probability

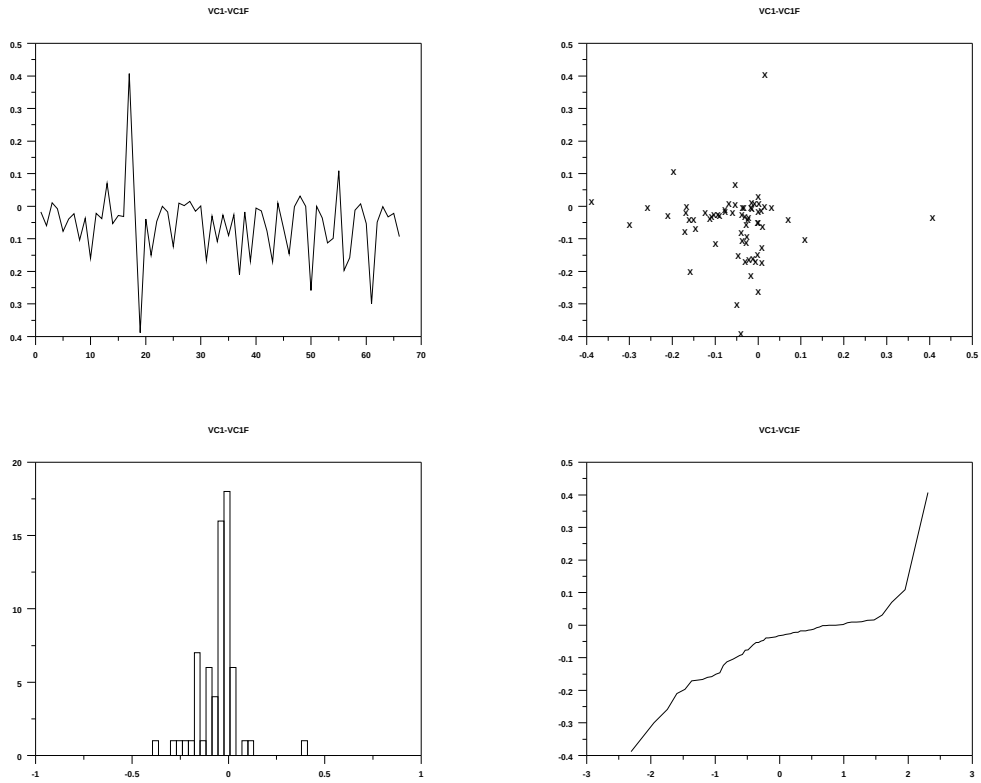

Figure 4.2. Overview plots describing D_i: order plot, lag plot, histogram, normal probability (quantile) plot

plots for visual inspection: figure 4.2 shows some overview plots of the pair differences. The assumption for a matched pair t-test is that this variable has a normal distribution. The histogram shows that the data is skewed and has some outliers. The non-normality can also be seen in the probability plot (a normal distribution would give a straight line). We applied three quantitative tests for normality, supported by the GENSTAT statistical software package (Anderson-Darling, Cramer von Mises, Watson), all tests confirmed that the data is non-normal at a 1% level. If we would ignore the fact that the normality assumption is not met, we would find a significant difference at the $p < 0.001$ level. We repeated the Anderson-Darling test for the 66 possible pairs of systems in a set of 12 related systems; 17 pairs proved to be normal according to the Anderson-Darling

test, the rest was not normal. We think this fact might be due to outliers: for some of the topics there are only three relevant documents, making the average precision values very sensitive to changes. A larger topic set will probably show more 'normal' pairs. The experiment shows however that the paired t-test is not always usable for IR data. The following sections will address non-parametric alternatives.

4.4.2.2. *Wilcoxon signed-rank test.* The Wilcoxon signed-rank test is defined as follows: given n pairs of observations, the absolute difference $|D_i|$ is calculated for each pair. Subsequently, ranks are assigned to these differences in ascending order, i.e. the smallest absolute difference receives rank 1, the one but smallest rank 2 etc. Subsequently each rank is prefixed with the sign of the difference and two sums are computed, one for the positive ranks and one for the negative ranks (W^+ and W^-). The intuition is that when the samples come from the same distribution, W^+ and W^- will hardly differ. If the test statistic exceeds the critical value, the null hypothesis must be rejected. However, it is unknown in what way the distributions differ. Usually an experimenter wants to make statements about differences in means. An additional assumption about equality of distribution parameters is necessary to allow for these inferences. The assumptions for the Wilcoxon test are:

(1) We take $D_i = \delta + e_i$, thus each difference consists of a constant and a random error term. For the null hypothesis that both samples come from a similar distribution $\delta = 0$
(2) The distribution of e_i is symmetric. This means that the median is equal to the means of the distribution.
(3) The e_i's are mutually independent (each query is independent from the other queries).
(4) The measurement scale of the D_i is at least interval[11].

The assumptions are thus less strict than for a t-test because only a symmetric distribution is assumed and not a normal distribution.

In a discussion of the Wilcoxon test, Savoy (1997) argues that care has to be taken to define when measurements really differ. Two observations that only differ in the third or fourth decimal obviously should be classified as tied, effectively reducing the power of the test (the same argument holds for the sign test).

As an illustration we show a data summary of the TRU retrieval system equipped with 12 different conflation modules (cf. chapter 6). The overview has been produced by GENSTAT and includes markers for skewed (non-symmetric) distributions. Out of 12 conflation modules, 3 modules produce a skewed distribution. This means that the applicability of the Wilcoxon is limited.

4.4.2.3. *Sign test.* Another alternative for the paired t-test is the sign test for matched pairs, which does not assume a symmetric distribution. The sign test just uses the information whether one of the scores in a pair is higher (+) or lower (-) than the other

[11]In statistics, four types of measurement scale, which define how to interpret numerical data. The most simple scale is *nominal* scale, where numbers are just arbitrary labels, numerical shorthands for textual descriptions. An *ordinal* scale assigns an order to numbers. An *interval* scale is an ordinal scale where am equivalent difference between two arbitrary numbers from the scale reflects an equivalent difference in the real world (e.g. the Celsius scale for temperature). The *ratio* scale is an interval scale where equivalent ratios taken at arbitrary points from the scale can also be equally interpreted.

Identifier	Minimum	Mean	Maximum	Values	Missing	
vc1ow	0.0010	0.2916	0.9604	66	0	
vc2f	0.0255	0.2746	0.9659	66	0	
vc2ow	0.0164	0.3037	0.9726	66	0	
vc4fow	0.0117	0.3211	0.9705	66	0	
vc1	0.0027	0.2195	0.9553	66	0	Skew
vc2	0.0167	0.2236	0.9659	66	0	Skew
vc4f	0.0151	0.2901	0.9691	66	0	
vc4ow	0.0161	0.3186	0.9705	66	0	
vc4	0.0176	0.2368	0.9691	66	0	Skew
vc1fow	0.0013	0.3015	0.9604	66	0	
vc1f	0.0023	0.2732	0.9553	66	0	
vc2fow	0.0000	0.2862	0.9000	66	0	

Table 4.2. GENSTAT data summary of 12 retrieval runs

score. If both samples come from an identical distribution, we would expect an almost equal number of pluses and minuses. The expected distribution of the sum of pluses can be described by a binomial distribution with $p(+) = p(-) = 0.5$. The assumptions for this test are:

(1) $D_i = \theta + e_i$
(2) The e_i's are mutually independent (each query is independent from the other queries).
(3) e_i are observations from a continuous population with median 0.
(4) The measurement scale is at least ordinal within each pair.
(5) The pairs are internally consistent, i.e. the projection of a performance difference is consistent for all pairs.

The null hypothesis we would like to test is:

$$H_0 : p(+) = p(-)$$

i.e. both samples are derived from populations with the same median. When the number of observations is large enough, one can use the normal distribution as a good approximation of the binomial distribution.

We will apply the sign test on the same data as in section 4.4.2.1: The table shows

Two-sample sign Test		
Variate	Size	Median
vc1	66	0.1389
vc1f	66	0.2167
Test if difference of medians equals 0		
	Test statistic:	12
	Effective sample size:	63
	Two-sided probability level:	0.000

Table 4.3. GENSTAT output for sign test

that after removing ties, vc1 is better than vc1f in 12 of 63 cases, which means that vc1f is better than vc1 in 51 cases. Common sense would suggest that vc1 is the better system. Indeed, the sign test detects a significant difference between vc1 and vc1f with great confidence: the p-value is smaller than 0.000. However, we cannot say anything about a confidence interval, or estimate the size of the difference between means or median, because the absolute value of the differences is ignored in the sign test. If the distributions are not symmetric, the best interpretation of a significant sign test is that the difference between the medians of the distribution is not equal to zero.

4.4.2.4. *Paired tests: conclusions.* Most researchers claim that, strictly speaking, only the sign test can be applied to IR measurement data. The disadvantage of using the sign test is that the method has a low power. A second disadvantage of non-parametric tests is that it is less straightforward to compute confidence intervals because the methods start from rank data and ignore absolute differences. We think a paired t-test should not be dismissed a priori; in some cases the distribution of pair differences is close to normal and then a t-test is to be preferred because of its higher power.

4.4.3. Comparison of more than two distributions. When we want to compare more than two IR systems, the naive approach would be to apply the techniques we discussed in the previous section in a serial fashion, as a sequence of independent tests.

We know that when we only compare two runs with a test at $\alpha = 0.05$ level, then the probability that we correctly conclude that a difference is significant is at least $(1 - \alpha) = 0.95$. Now suppose we want to compare m runs. In that case we have to perform $\binom{m}{2} = m(m-1)/2$ tests between all possible pairs. If we assume that these tests are independent, the probability that we do not make any mistake is $(1-\alpha)^{(m(m-1)/2)}$. For $n = 10$ and $\alpha = 0.05$ this results in a very small probability: 0.099.

A solution to this problem is to construct an integrated model for all data, instead of regarding all comparisons as independent tests, which allows for a more careful modelling of effects, interactions and error terms. The usual approach is to model the data using a linear model, which will be discussed in subsection 4.4.3.1. Test procedures have been developed for these linear models which are especially designed to control the total α. Just like the tests for paired samples, there are parametric and non-parametric approaches. Because both approaches are applied by IR researchers, we will briefly discuss each of them. The usual approach for tests based on the linear model is that the experimenter will first test globally whether there are any significant differences and, if this is the case, will subsequently test which pairs of systems are significantly different. The tests in the second step are called *multiple comparison tests*. We will first introduce the linear model, the analysis of variance. After the general introduction on the comparison of several means we will devote a separate section to the application of this kind of tests to IR data, which comes not without problems. Subsequently we will discuss a non-parametric alternative: the Friedman test.

4.4.3.1. *The General Linear Model.* In section 4.4.1.1 we introduced the notion of hypothesis testing in an informal manner. In comparing two hypotheses in fact two models of the data are compared. The most common models are *linear models* i.e. models in which an observation is taken to be a linear combination of several effects. Suppose we

denote an observation on a dependent variable of interest as Y_i, then we can account for this observation with a linear model:

$$Y_i = \beta_0 X_{0_i} + \beta_1 X_{1_i} + \beta_2 X_{2_i} + \beta_n X_{n_i} + \epsilon_i \tag{55}$$

Here, $\beta_0 X_0$ represents the influence of constant factors (usually equivalent to the mean of the sample) , $X_1 \cdots X_n$ are binary indicator variables which indicate whether an observation is part of group n (for each i only one of the indicator variables equals 1). The groups correspond with different levels or different types of the controlled variable X, often called treatment or factor. An example factor could be the type of stemming algorithm used in a system. ϵ is the residual error, denoting the random variation due to chance. The goal of the experiment is to estimate the betas, which describe the relationship between a factor and the dependent variable.

4.4.3.2. *Comparing Linear Models: ANOVA.* Rather than presenting hypothesis tests as a comparison between "between group" and "within group" variability, we prefer the model comparison view presented in (Maxwell & Delaney, 1990).

The basic idea for a model comparison based hypothesis test, is that both H_1 and H_0 represent linear models. H_1 is the full model corresponding to our intuitions, and H_0 is a more restrictive model with fewer parameters, corresponding to the idea that the parameter(s) we are investigating has no effect on the dependent parameter: the H_0 hypothesis. We want to know whether the full model can describe the data in a more adequate way, normalised by a factor denoting the simplicity of the model. In other words: does the full model fit the data better than what would be expected given the fact that it contains more parameters and thus by definition has a better fit of the data? The fit of a model is measured by the error term E_R[12]for the restricted models, and E_F for the full model. The adequacy of the models can be compared by looking at the proportional increase in error (PIE), going from the full to the restricted model:

$$\text{PIE} = \frac{E_R - E_F}{E_F} \tag{56}$$

A good measure for simplicity/complexity of a model is the *degrees of freedom (df)* parameter. The df parameter is defined as the number of independent observations minus the number of independent parameters estimated in an experiment. The normalised PIE ratio:

$$F = \frac{(E_R - E_F)/(\text{df}_R - \text{df}_F)}{E_F/\text{df}_F} \tag{57}$$

is better known as the F-ratio. If we assume that the error terms ϵ_i are normally distributed with a zero mean, the F ratio follows the F distribution, one of the standard statistical distributions. Thus the F-ratio can be used to do well-founded hypothesis tests for the comparisons of two linear models.

Significance tests based on this F-test are called analysis of variance (ANOVA). Most common are experiments with one or two controlled variables, which can be handled by a one-way and two-way ANOVA respectively. Including more factors in the same

[12]The error term is defined as the sum of squares $\sum_j \sum_i e_i^2$.

experiment makes it possible to model the interaction between factors, complicating the analysis at the same time.

Given a certain model, the experimenter can choose to do one observation per subject, i.e. a "between subjects" design or to test different levels of a parameter on the same subject: a "within subjects" design[13], in a way an analogue of the paired tests discussed in section 4.4.2.1. Care has to be taken to choose the correct error terms in the F-test, which is a matter of choosing a model which is appropriate for the data.

After introducing the notion of linear models we will now continue with a discussion of several methods for multiple comparison tests. In section 4.4.3.4 we will discuss the application of the analysis of variance to IR experiments.

4.4.3.3. *Multiple Comparison tests.* If the F-test of an analysis of variance has led us to reject the null hypothesis that all samples come from the same distribution, we now have to find out where the real differences are. Choosing a method for making comparisons between multiple means is a quite complicated and even a bit controversial issue. It has been the subject of heated debates between different camps in the statistic community. As often in these "religious" debates, there is no absolute truth. The background of the debate is the trade-off between Type I and Type II errors. On the one hand there are conservative experimenters that prefer a low Type I error, on the other hand there are more pragmatic statisticians that focus on a low Type II error, because otherwise one would never detect any significant differences.

Following (Maxwell & Delaney, 1990) we define α_{PC} as the type I error per comparison and α_{EW} as the *experiment wise* type I error, i.e. the probability that we falsely conclude a significant difference at least once. As we have shown in section 4.4.3, the α_{EW} grows exponentially with the number of means which has to be compared: Because

m	α_{EW}
2	0.098
4	0.22
6	0.54
8	0.76
10	0.90

Table 4.4. α_{EW} as a function of m

the tests are dependent, these are even lower bounds! One can also calculate the average number of type I errors in the total experiment, this is simply $\alpha_{PC} * m(m-1)/2$. One obvious way to control α_{EW} in a situation with C comparisons, is to make use of the Bonferroni inequality:

$$1 - (1-\alpha)^C \leq C\alpha \tag{58}$$

If we choose $\alpha_{PC} = 0.05/C$ then it follows from (58) that $\alpha_{EW} \leq 0.05$ (Maxwell & Delaney, 1990, p.177). However, this method does not really help us in a situation where C is large, because this would require α_{PC} to be extremely small, thereby severely deteriorating the power of the test, which is equivalent to increasing the type II error. Even a conservative

[13]Sometimes also called a repeated measurements design.

statistician is bound to make some Type I errors during his professional life. Suppose he performs 500 independent hypothesis tests during his professional career where the null hypothesis is rejected. If each individual test was performed at a $\alpha = 0.05$ level, probably 25 of the 500 positive conclusions are false!

An example of an IR study which falls into this trap is Zobel (1998). The significance tests in this study do not take into account any global α_{EW}. The authors perform 7320 pairwise tests with three different types of tests: t-test, ANOVA (on a 2 sample set!) and Wilcoxon. They took $\alpha_{PC} = 0.5$, the tests yielded 3810 significantly differing pairs using the t-test. Thus, 191 of these cases are false positives!

There are a couple of standard approaches to do multiple comparisons:

Planned Comparisons: The experimenter can plan beforehand which comparisons he wants to make in order to validate his experimental hypotheses. This helps, because the number of planned comparisons can be restricted, which does not hurt the power of the test too much.

Fisher's protected LSD test: This is the oldest test, which is still popular because it is simple. The idea is to first test the null hypothesis that all samples have the same distribution (the omnibus test). This test has the goal to protect the experiment against a high α_{EW}. The idea is that we only do pairwise comparisons when the omnibus test shows that there is a significant difference. This test (an ANOVA) can in theory protect at an $\alpha_{EW} = 0.05$. The approach is attractive because α_{PC} does not have to be adjusted, because the experiment is 'protected' by the global F-test.

If the omnibus null hypothesis is rejected, the experimenter can proceed with pairwise comparisons. We can do a simultaneous comparison of all the means by computing the least significant difference (LSD). The LSD can be computed by the following formula:

$$LSD = t_{(\alpha=0.05/2,\nu=n)} \cdot \text{s.e.d.} \tag{59}$$

where n is the sample size and s.e.d. is the standard error of difference:

$$\text{s.e.d.} = \sqrt{\text{MSW}_A/n + \text{MSW}_B/n} = \sqrt{2\text{MSW}/n} \tag{60}$$

The s.e.d in formula (59) is based on a pooled estimate, and thus assumes equal variances for all means; therefore $\text{MSW}_A = \text{MSW}_B$[14].

An experimenter has to take care though to meet the assumptions of the test. Suppose there are some quite similar systems and one rather different system, then the omnibus null hypothesis will probably be rejected because the pooled error is relatively small, but the procedure will not really help us to control α_{EW}. So, the experimenter has to make sure that the data has homogeneous error variances, before applying the protected LSD.

Tukey's HSD test: Tukey (1953) designed a test to overcome the weakness of the protected LSD test. The test is based on the computation of the *honestly significant difference*[15], which serves to compare m means simultaneously, while controlling a global α_{EW}. The idea is simple, the null hypothesis assumes that

[14] MSW stands for Mean of Squares Within, an estimate for the variance within groups.

[15] sometimes also referred to as highest significant difference

all samples are taken from the same distribution. Now we compute the largest difference between two means that we could expect under this hypothesis at an α level of 0.05. If a difference between means exceeds this HSD, we can conclude that the difference is significant, while controlling α_{EW}. The HSD is based on the studentized range statistic Q:

$$HSD = D_i > Q(m, (n-1)(m-1), \alpha) \cdot \sqrt{MSE/n} \tag{61}$$

where MSE is the mean squared error term. The main difference with the protected LSD test is thus that α_{EW} is directly controlled.

The principal multiple comparison methods have been evaluated with Monte Carlo methods (Snedecor & Cochran, 1980, p.235). The conclusion of these studies is that the preferred method is dependent on the particular dataset and the relative cost of Type I and Type II errors. Given a choice, Snedecor & Cochran opt for the protected LSD method:

> On balance, Carmer and Swanson like the protected LSD, which has good control of Type I errors, more power than the Newman Keuls, studentized range (Tukey HSD) and Scheffe methods, and is easy to use.

An elaborate power simulation study by Hull et al. (1999) confirms the conclusion that the LSD method is to be preferred in most practical cases, because its power is much higher compared to other multiple comparison tests. This study however shows that 50 topics are not enough for a sensitive (Type II error < 0.1) significant test of a 0.05 absolute difference. The use of 100 topics already gives a much better sensitivity.

A paradoxical complication of any experiment involving the comparison of several means is that the minimum significant difference between systems is inversely related to the number of means which are being compared. The paradox lies in the fact that if we are primarily interested in comparing systems A and B, and add 10 others systems in the experiment, it is much harder to detect a significant difference between A and B. Note that a way out of this paradox is to realize that the "between two" and "between twelve" experiments pose very different questions. If one is interested in comparing A and B in the first place, one should simply compare just the two systems.

General conclusion is, that it is desirable to limit the number of comparisons to reduce the number of Type I and II errors. If there are a large number of comparisons, the LSD method has the highest power. Care has to be taken to control the global error rate, and to check that the protection is not invalidated because there is an odd system in the comparison.

One way to limit multiple comparisons is to plan experiments carefully with clear hypotheses. If there is no clear expectation about the effects of interest, one could start with a first series of exploratory experiments and conduct a second independent experiment to confirm effects found in the first experiment. This second experiment can focus on a small number of effects, decreasing the problems with a high error rate for the multiple comparison tests.

4.4.3.4. *Applying ANOVA to IR.* After the general introduction on the analysis of variance and the closely related multiple comparison tests, we will assess if and how the ANOVA test can be used for IR data.

Fortunately the type of research questions underlying batch IR experiments are simple: usually the question is whether a certain IR system is significantly better or worse than other systems; interaction effects are assumed to be non-existent. So the system variants are modeled as different levels of the factor system. However, as we mentioned earlier, the across-query variation is much larger than the across-system variation. Suppose we would compare 3 systems, and each system is tested on the same set of 50 queries. Then the error term would be almost completely determined by the across-query variation, so we would be unable to detect significant differences between systems even if they exist, because the samples are small in comparison to the between query variance. Because it is very costly to extend the test collection size (more in particular, the set of relevance judgements), the best solution is to test all systems on the full query set, a completely crossed design, and analyse the data as a within-subjects experiment. Such an approach maximises experimental power given a limited number of subjects. This means that the query is treated as a second factor in the model:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \tag{62}$$

where Y_{ij} represents the score (e.g. average precision) for system variant i ($i = 1, ..., m$) and query j ($j = 1, ..., n$), μ is the overall mean score, α is the system version effect, β is the query effect and ϵ represents the random variation about the mean. The experiment is a comparison of m systems tested on n topics.

This model is also referred to as a mixed model without replication. There is one factor with fixed levels and one factor which is sampled. A peculiarity of the model is that there is only one observation per cell (system/query pair) because an IR system is deterministic (Buckley & Voorhees, 2000). Thus we cannot estimate the sample variance per cell. This has as a consequence that we cannot estimate a completely general interaction model (so we simply assume that there is no interaction) and can only test on system effects. A different perspective is to say that the error term is equivalent to the interaction term (Maxwell & Delaney, 1990, p. 431–432). Main point is, however, that in theory a two way ANOVA allows the IR experimenter to reach more precise results with the same amount of queries, because the influence of the query on retrieval performance is explicitly modelled. In the so-called 'query track' at TREC-8 and TREC-9, the experimental design was different. For every topic, different queries were produced. Thus for each topic-system combination, there are several observations, which would make it possible to investigate the interaction between systems and topics. This study has not been carried out yet. Until now, the only study based on the query track test collection had the target to investigate the robustness of different performance measures (Buckley & Voorhees, 2000).

The two-way mixed model analysis for TREC-style IR experiments has been advocated by Tague-Sutcliffe (Tague-Sutcliffe & Blustein, 1995) in order to by-pass the problem of between query variance while controlling α_{EW} at the same time. However, the use of ANOVA for the analysis of IR experiments has also been criticised because IR data sets do not meet the parametric assumptions. The assumptions for a valid F-test

are: the error distributions are normal and independent and they have equal variances. In fact none of these assumptions are fully met, but usually ANOVA is quite robust to deviations from the normality assumptions, especially if the sample size is considerable (sometimes the figure 30 is mentioned as a lower bound (Hayes, 1981)). A possibility to get around the problem of non-normal data or data with unequal variances is to apply transformations on the data in order to stabilise the variance and apply ANOVA on the transformed data. For example, one could apply square root, log or arcsine transformation to produce less skewed data (Maxwell & Delaney, 1990, p.112). Tague-Sutcliffe applied arcsine transformation in her TREC-3 study. She compared 42 runs by applying a standard ANOVA, an ANOVA on arcsine transformed data and a non-parametric ANOVA variant: the Friedman test (cf. 4.4.3.6). The overall F-test showed significance differences in all three cases, the subsequent Scheffé[16] multiple comparison tests showed that there were very few differences in the equivalence groupings.

In a small scale experiment on the UPLIFT collection (cf.section C) we checked the normality assumption and the effectiveness of data transformations. We performed an analysis of variance on the average precision figures produced by the same set of twelve systems as presented in 4.4.2.2 and we subsequently did the same analysis on three different transformations of the data: log, square root and arcsine transformation. The twelve systems are all minor variations of each other, so one would expect the error variances to be homogeneous. Figure 4.3 shows some plots describing the residual distribution of the model fitted on the original data. Figure 4.4 shows plots for the residuals of the model for the square root transformed data. The plot shows that the residual distribution for the model fitted on the original data is not quite normal, but after applying the square root transformation the fitted value plot is much more homogeneous. A plot of the means and the standard error of difference is shown in the figures 4.5 and 4.6. We also reproduce the ANOVA table of both analyses below: It's clear from the tables

Analysis of variance					
Variate: y					
Source of variation	d.f.	s.s.	m.s.	v.r.	F pr.
pers stratum	65	27.805283	0.427774	78.69	
pers.*Units* stratum					
method	11	0.924631	0.084057	15.46	<.001
Residual	714	3.881476	0.005436		
Total	790	32.611391			
Standard errors of differences of means					
Table	method				
rep.	unequal				
d.f.	714				
s.e.d	0.01288	max-min			

Table 4.5. Analysis of variance (GENSTAT) for the untransformed data

and figures that the square root transformation does not really change the analysis. Both

[16]The Scheffé multiple comparison test is another type of MCT, which controls the total α_{EW}.

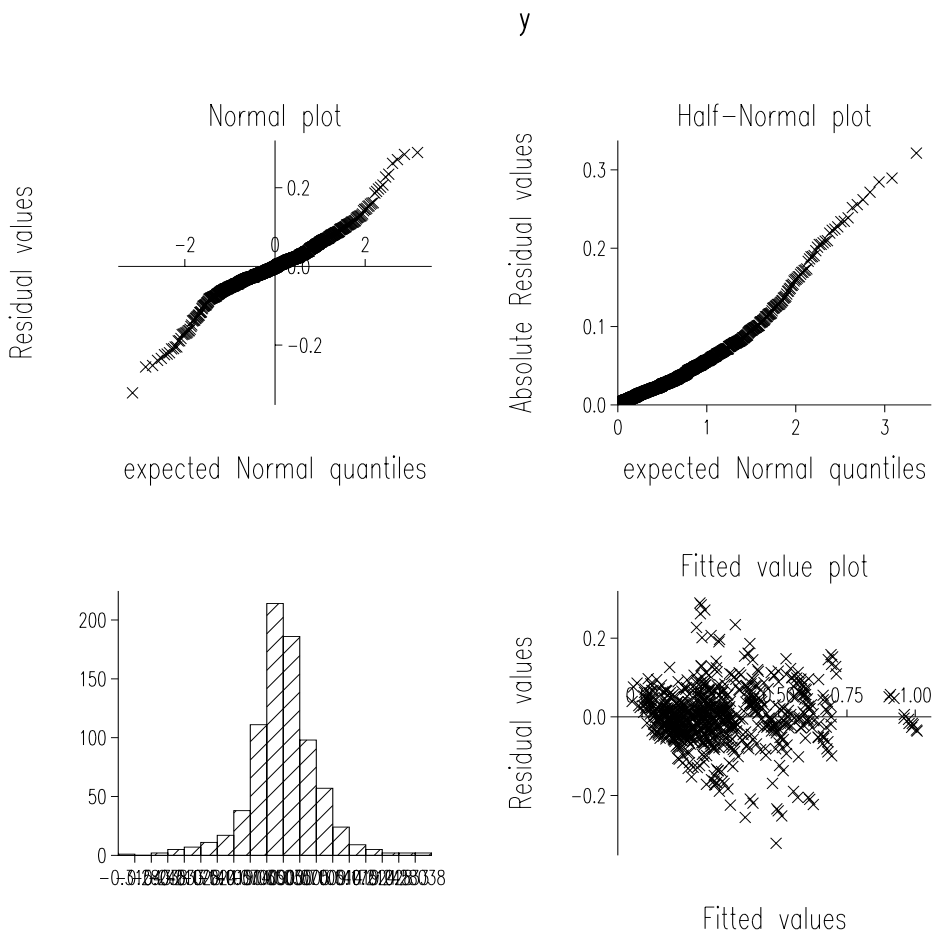

Figure 4.3. Overview plots describing the residual distribution of the original data: normal plot, half normal plot, histogram, fitted values plot

ANOVA's show an effect (F is significant) and pairwise comparisons yield the same order and grouping of systems. Therefore there are indications to conclude that although the residuals have no exact normal distribution, the ANOVA on the untransformed data gives reliable results, at least reliable enough for our purpose.

There are some disadvantages to working with transformed data: first of all it is hard to interpret the transformed values of average precision. Another reason to use these transformations with some reservation is the fact that if a null hypothesis about the transformed data is rejected this has, strictly speaking, no implication on the analysis of the original data.

Besides Tague-Sutcliffe's TREC-3 study, the analysis of variance has been applied to IR problems on a regular basis. Gordon & Pathak (1999) describe a study where 8 Web

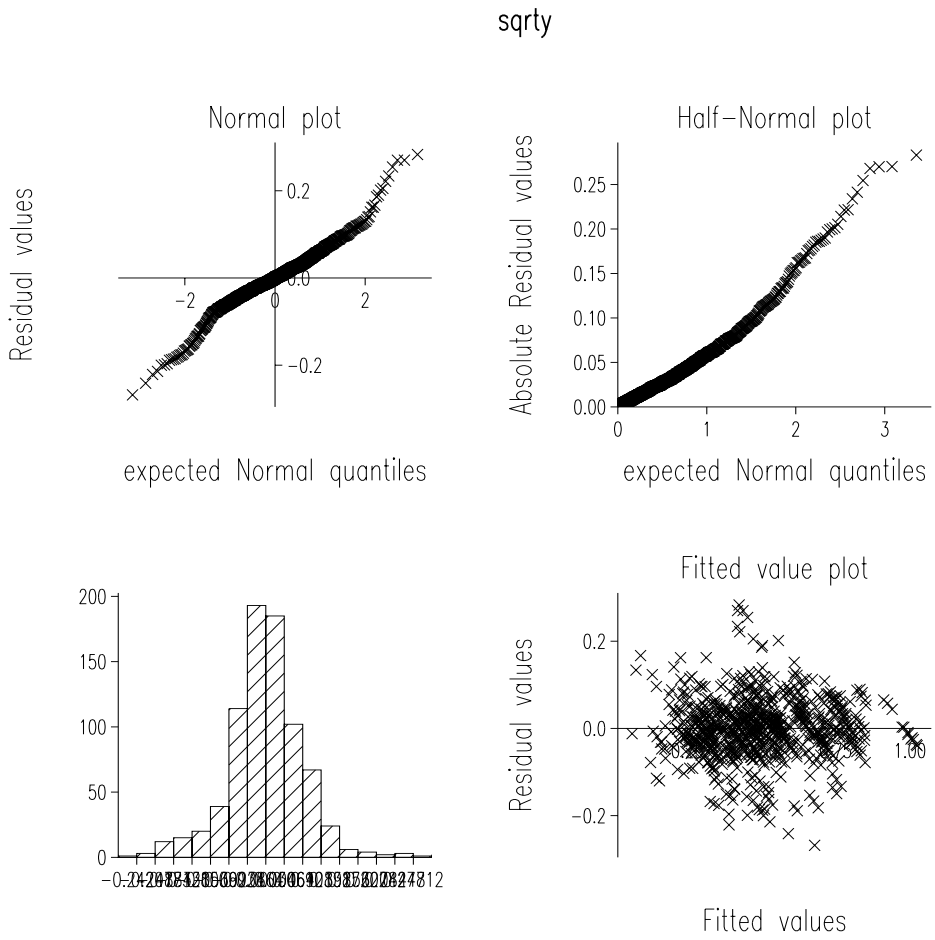

Figure 4.4. Overview plots describing the residual distribution of the square root transformed data: normal plot, half normal plot, histogram, fitted values plot

search engines are compared on a test collection of 33 queries. The authors claim that this number of queries is enough to satisfy the normality assumptions. The systems are compared using Tukey HSD, without any checks on homogeneity of variances. The interactive track at TREC probably forms a safer area for ANOVA tests because the data shows natural variation. Hersh et al. (2000) report on interactive track work: 24 subjects had to do a search task based on 6 topics. For 3 topics the back-end of the system used Okapi ranking, for the 3 other topics, it was based on *tf.idf*. The tasks were assigned to the subjects in a controlled way, to account for order affects. This set-up seems perfect for a two-way ANOVA. Because there were 24 observations per cell, the experimenters could also estimate the interaction between topic and system. The result of the ANOVA is that

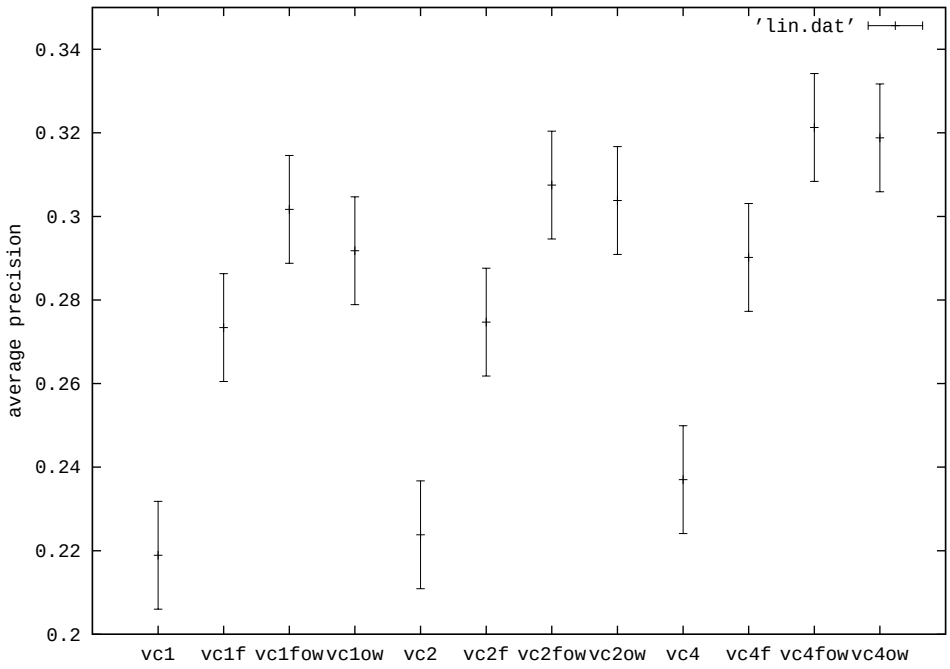

Figure 4.5. Mean average precision and errorbars of 12 different systems. When errorbars overlap, systems are not significantly different.

Analysis of variance					
Variate: $\sqrt{y}$					
Source of variation	d.f.	s.s.	m.s.	v.r.	F pr.
pers stratum	65	23.801823	0.366182	68.95	
pers.*Units* stratum					
method	11	1.029348	0.093577	17.62	<.001
Residual	714	3.791901	0.005311		
Total	790	28.623071			

Standard errors of differences of means		
Table	method	
rep.	unequal	
d.f.	714	
s.e.d.	0.01278X	min.rep
	0.01273	max-min
	0.01269	max.rep

Table 4.6. Analysis of variance (GENSTAT) for the square root transformed data

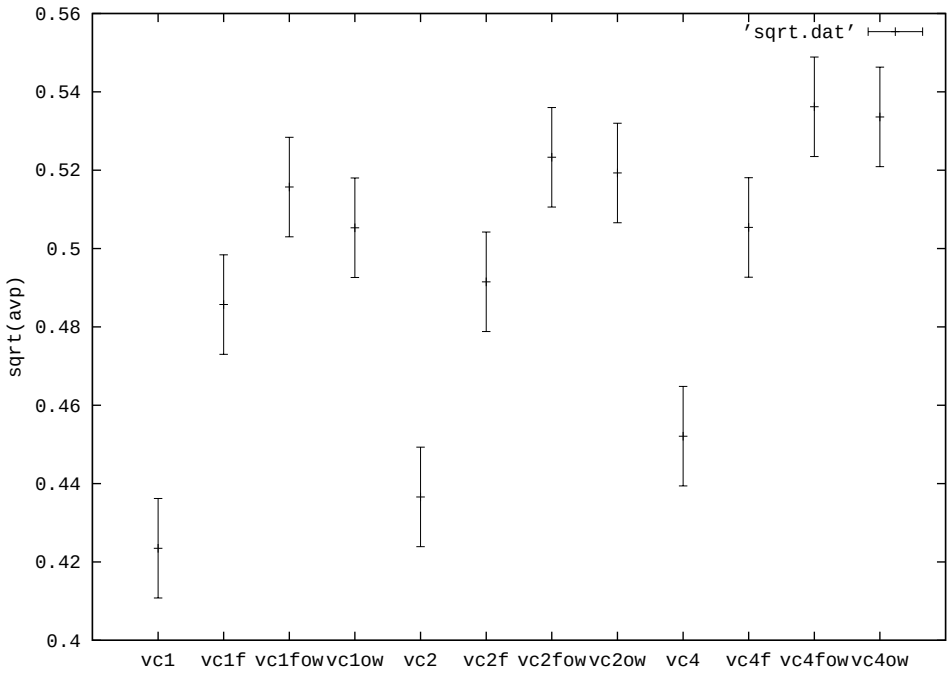

Figure 4.6. Mean average precision and errorbars of 12 different systems. When errorbars overlap, systems are not significantly different.

there is no difference between the two back-ends when they are used in an interactive setting, while a simple t-test indicated a significant difference.

4.4.3.5. *Multiple comparison tests in a within-subjects design.* We have already argued that an unbalanced set of systems (where there is one odd system between a number of quite similar systems) invalidates the overall F-test, because it violates the homogeneity of error variances assumption. In order to carry out classical comparison tests for contrasts, like the Tukey test in a within subjects design, the data also has to respect the so-called *sphericity* assumption (Maxwell & Delaney, 1990, p.471), which basically means that for any system pair (i, j) the variance of the difference between paired samples should be equal to the variance of any other pair (i, j). It is unlikely that this assumption is met for ordinary IR experiments. This does not mean that analysis of variance is completely disqualified, but that the probability of a type I error is higher than specified in the case of unmet assumptions (Maxwell & Delaney, 1990, p.471).

A possible alternative to a mixed model univariate analysis is a multivariate approach (Maxwell & Delaney, 1990, p.605) or a multilevel approach (Bryk & Raudenbusch, 1992). Because these set-ups breakup the data in smaller groups in order to get multiple observations per cell, we need far more observations (queries) to create a powerful test. Instead we will discuss a non-parametric alternative version of the analysis of variance which has successfully applied in several IR studies in the next section.

4.4.3.6. *Friedman test.* The more assumptions we can make about our data, the more powerful tests we can apply. Seen from this perspective, an experimenter should always prefer to apply parametric tests, given the fact that the costs of building a test collection are linearly dependent on the number of topics. However, some researchers question the validity of the normality assumption associated with parametric tests. Hull proposes the non-parametric Friedman test as an alternative (Hull, 1993).

The Friedman test can be considered as an extension to the sign test for matched n-tuples (n treatments). This means that for $n = 2$ the Friedman test is equivalent to the sign test. Just like the Wilcoxon signed rank test, this test is based on relative ranks instead of the original data. The assumptions for this test are:

(1) the m n-variate random variables are mutually independent. (The results within one block (=topic) do not influence the results within the other block.)
(2) Within each block the observations can be ranked according to some criterion of interest.

The idea underlying the Friedman test is that each ranking of the m variables within a block is equally likely, which corresponds to a zero treatment effect. The test can be regarded as an analysis of variance on ranks. The procedure consists of two steps:

(1) Compute the test statistic $T = \frac{(n-1)[B-nm(m+1)^2/4]}{A-B}$ where $A = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}^2$, $B = 1/b\sum_{j=1}^{n} R_{.j}^2$ and $R_{.j} = \sum_{i=1}^{n} R_{ij}$
(2) If the null hypothesis is rejected we can apply multiple comparison tests based on the test statistic $|R_k - R_l| > t_{1-\alpha/2}\sqrt{[\frac{2n(A-B)}{(n-1)(m-1)}]}$. This is a non-parametric version of the LSD test.

For a detailed account of Friedman tests we refer to Conover (1980).

As usual with statistical tests, the Friedman test also has some disadvantages. Like the Wilcoxon and sign test, the Friedman test can only give clarity on the question which systems are significantly different. But because the test is applied on transformed data, the relationship with the original scores is a bit obscured. An more serious problem is that the Friedman test is quite sensitive to the composition of the test set, taking out a run from a test set can potentially change the order of the remaining runs. An example where the overall rank order of systems A and B is reversed is shown in tables 4.7 and 4.8. An interesting property of the test is that there is automatic normalisation over

System	ranks						sum ranks	overall rank
A	1	2	1	1	3	2	10	1
B	2	1	3	3	1	1	11	2
C	3	3	2	2	4	4	18	3
D	4	4	4	4	2	3	21	4

Table 4.7. Rank table of systems A,B,C and D

queries, because only rank differences are taken into account, not the absolute values.

An example study of good statistical practice is Leighton & Srivastava (1999). In a comparison study of web search engines the Friedman test was used because a normality

System	ranks						sum ranks	overall rank
A	1	2	1	1	3	2	10	2
B	2	1	2	2	1	1	9	1
D	3	3	3	3	2	3	17	3

Table 4.8. Rank table of systems A,B and D

test revealed that the residual distribution was not normal. The Friedman test was based on medians instead of means because the data was skewed. This has the advantage that the results are less sensitive to outliers. Another example of an application of the Friedman test on IR data is Kekäläinen & Järvelin (2000).

4.4.3.7. *Multiple comparisons: conclusions.* Our presentation of techniques for the comparison of more than 2 runs has shown that the choice of a particular multiple comparison technique is a rather complex and on some points even controversial matter. The basic problem underlying the controversy is the fact that there is a trade-off between Type-I and Type-II error. It is a matter of choice which type of error is more "expensive". There is some analogy here with the notion of cost function which is used to optimise the decisions of a classifier. Usually a Type I error is seen as a more serious error. Therefore techniques have been developed to control the α_{EW}, the probability to make an error while making pairwise comparisons between several systems. We have a light preference for applying the simple protected LSD method, because power tests have shown that it has superior power. Application of ANOVA for IR experiments has been criticised because residual error distributions are not normal (Salton & McGill, 1983). Salton's suggestion to use sign tests is prone to a high α_{EW}. However, transformations can help to stabilise error variances. Analysis of variance of the completely crossed data set of an IR experiment presents even more problems. In theory, the mixed model analysis is capable to remove the effect of the topic variance from the error term. However, a mixed model analysis has even more strict requirements on the residual variances. Moreover we cannot model interaction between systems and topics because we have only one observation per cell. We think that when the experimenter is careful in selecting systems for a comparison (to control the homogeneity of variance) the analysis of variance is still a helpful tool to interpret the significance of differences between means. However, one should not make bold conclusions when the $p-$value is close to the α level. The Friedman test is a good alternative with some minor disadvantages. Like for the ANOVA on transformed data, its conclusions are more difficult to interpret, because they are not on the original scale. We think that the best strategy is to be parsimonious with the selection of systems for a test and to create test sets in a balanced manner: either really different systems or one system with several minor variants should be selected for a test.

4.4.4. Discussion. In this subsection we will discuss several viewpoints of IR researchers on the utility and applicability of significance (tests) for IR experiments. First we discuss related studies of significance tests for IR, subsequently we discuss the notion of "practical significance", which is commonly used. Subsequently we discuss in which cases it is not useful to run statistical significance tests and conclude with formulating the approach taken for the experiments in chapter 5 and 6.

Related work. The issue of selecting tests for statistical significance testing is rather controversial among IR researchers. Maybe partly due to this lack of agreement most researchers avoid statistical tests and work with heuristics[17] to determine the significance of performance differences. Application of statistical tests is not without problems indeed. The assumptions made by the the more powerful tests are usually not fully met by the experimental data. The tests with less stringent assumptions tend to be less powerful or more difficult to interpret. Some researchers have a conservative viewpoint: given the fact that IR performance measures mostly do not conform to the normal distribution, only the non-parametric sign test is allowed (Rijsbergen, 1979) and (Salton & McGill, 1983). Other researchers like Tague-Sutcliffe & Blustein (1995) and Hull et al. (1999) are more pragmatic and also apply more powerful tests like ANOVA, but warn the experimenter that the tests have to be interpreted with care. Some of the tests are known to be relatively robust against violations of assumptions. Savoy seems to follow both schools, as he presents test results using different tests and different collections (Savoy, 1997). He proposes to use the bootstrap method (cf. section 4.4.1.3) as an alternative to classical parametric and non-parametric tests because it is assumption free. However, this method is more complicated. In his study, the bootstrap method yields similar results as the sign test.

Practical significance. In addition to the question whether statistical significance tests can be applied to IR data, there is of course the question whether a certain *statistically significant* performance difference has any *practical significance*. (Sparck Jones, 1974) proposes the following rule of thumb: compute the absolute difference δ between two values of a performance measure. If $\delta > 10\%$ one can speak of a *material* difference, if $5\% < \delta < 10\%$ the difference is *noticeable* and if $\delta < 5\%$ the difference is *not noticeable*. Many authors use this rule of thumb when discussing their results (e.g. Burgin, 1992). However, strictly speaking, a material difference can be statistically not significant and a not noticeable difference can be statistically significant.

When are significance tests meaningful? Until now the focus has been on the discussion of different types of significance tests rather than the issue in which cases we want to apply these tests. There are many different types of IR experiments possible, but not every type of IR experiment merits a significance test. Here are some example experiments:

(1) Comparison of different retrieval models
(2) Comparison of a baseline model with an extension
(3) Comparison of two model extensions
(4) Finding proper parameter settings for a model
(5) Comparison of retrieval models across different test collections

Some of these example questions correspond to more fundamental and some to more detailed research questions, corresponding to refining a model or comparing basic models. The standard way to answer such a research question is to test systems on standard test collections. There is a danger though, especially for the experiments that aim to refine models or to find optimal parameters, that a model will be overtuned to the data. This pitfall is a classical problem in machine learning experiments. In the machine learning community, overtuning is usually prevented by training (tuning) a system on only

[17] We will discuss these later in this section.

a part of the data and testing the system on the remaining data. Often this process is randomised (n-fold cross validation) in order to smooth out accidental effects.

This approach can be followed to a certain extent in IR. Some retrieval models contain parameters that have to be tuned. The experimenter should take care not to tune parameters to a certain test collection, or more specifically, to a certain collection of topics. Of course it is viable to tune parameters to a certain document collection. A researcher must however realize that this limits the general applicability of the model, because the model has to be tuned for a new document collection, something which is not desirable for highly dynamic document collections.

An ideal IR experiment is a blind experiment with new data, i.e. with new topics and post-test relevance judgements. In this case, it is more difficult for an experimenter to tune on the test data. In TREC context, looking at the test topics is not forbidden, but if this leads to a change of the system, or system parameter, the experiment has to be classified as a manual run. That does not mean it is an uninteresting experiment. Manual runs can dramatically increase the quality of an IR experiment, by setting an upper bound for automatic systems and by improving the quality of the pool. TREC has proven to be a cost effective model to (i) perform blind tests on participating systems (ii) build test collections which can be used in later experiments. The latter result is extremely important for the IR field. One way to avoid overtuning to a particular test collection is to do a blind test on a separate test collection. A good example of a collection of test sets are the topic collections for the ad hoc task, developed during TREC-6, -7 and TREC-8. There are 3 sets of 50 topics with judgements on the same document collection.

Coming back to the question which type of research questions merit a thorough significance test procedure, we think that it is not useful to perform significance tests on systems that just differ in their parameter settings. For this kind of experiments it is much more interesting to check whether performance differences are of a systematic nature across topic collections or even across document collections. In that case, the parameter setting seems to capture a collection specific or language intrinsic phenomenon. Significance tests could help though to decide when it is useful to check across collections. The main use of significance tests is when the experimenter has a clear research question which is not a parameter optimisation problem. An example research question is to compare the Okapi model with the vector space model based on the Lnu.ltu formula.

Guidelines for sound inference from data. We take the position that the careful application of statistical tests can help the researcher to assess the strength of a certain effect. A good experiment starts with a clear statement about the intuition or theory which we want to test. A guideline here is to design simple contrastive experiments i.e. experiments where just one hypothesis is tested, using strictly additive models. More complex experiments would require far more data than which is generally available[18]. Of course it is not realistic to assume there is no interaction between effects, but it is probably the best that realistically can be achieved.

Subsequently, these ideas have to be implemented, the corresponding system versions have to be debugged and tested. Finally the real evaluation should preferably done

[18]The afore mentioned TREC6,7,8 collection with 150 topics might be an interesting collection to do more complex designs.

on a separate test collection. Because no significance test is ideal, it is recommendable to do several tests, taking basic precautions to ensure a reliable error term for the parametric tests. Restricting oneself to non-parametric methods like the sign test has the disadvantage that quantitative confidence intervals are not available.

In our experiments in chapter 5 and 6 we will use sign tests for all cases when we want to compare just two systems. When more systems have to be compared, or more than two system variants, it is better to apply ANOVA and/or Friedman because a series of pairwise t-tests has far higher type I error. When applying ANOVA, one has to take care that not to compare apples and oranges, i.e. dependent systems have to be removed from the comparison, or the comparison should be restricted to variant systems, in order to work with a correct error term. A comparison of some variant runs and a baseline run in a general linear model is not sound. One either has to remove the baseline run from the ANOVA or work with a more complicated nested design. An alternative or complementary test is the Friedman test, but we should be aware of its sensitivity to the composition of the test set. For the multiple comparison tests we prefer the LSD test, which has a good trade-off between type-I and type-II error (Hull et al., 1999). In our experiments we will use the Friedman test.

In general, it seems a good strategy to be conservative in drawing conclusions. Firstly assumptions of significance tests should be checked. Our experiments confirmed that data from IR experiments often does not meet these assumptions, especially in the case of parametric tests. If assumptions are met, an even more conservative strategy is to apply different types of significance tests or to run tests on different test collections and only draw conclusions when the results are equivocal.

4.5. Pool quality

In the previous section we already mentioned that IR systems are often evaluated on existing test collections. There are two potential problems with "re-using" a test collection: (i) it takes more discipline to perform a really blind experiment and extra care not to tune on the data (cf. section 4.4.4) (ii) post-hoc runs are unjudged runs by definition. An unjudged run is a run that did not contribute to the pool. For judged runs we know that at least the top 100 (the most common pool depth) is judged. For unjudged runs, this will not be the case. The percentage of judged documents (the judged fraction) will be lower. However, presenting results of unjudged runs is very common. Even at TREC not every run is judged. Participants can submit runs and because of the limited capacity and budget, the pool is based on a selection of the submitted runs, usually one run per participating site. For judged runs, the number of judged documents is 100, for unjudged runs this number is lower. That means that the calculated performance measures are more reliable for judged runs. The difference in reliability between judged and unjudged runs has been studied. Buckley has suggested the following experiment: recompute the average precision of every run that contributed to the pool based on a pool without the judged documents that were uniquely contributed by the very same run. Finally, compute the averages of the average differences or improvements in performance over the runs. Zobel (1998) ran this experiment on the TREC-5 run set and reported an average improvement of 0.5% over 61 runs with a maximum of 3.5%. The fact whether a run is judged or not thus seems to play a minor role in the TREC-5 dataset. However,

the TREC-7 CLIR evaluation showed a different picture: 14 runs were used to construct a multilingual pool (for English, French, German and Italian). Here the maximum difference was 0.0511, corresponding to a performance increase of 28% and an average difference of 0.02 (14%) (Kraaij et al., 2000). The latter figures indicate that a smaller pool is less reliable for unjudged runs, because a smaller pool is probably less diverse. A more important explanation for the lower reliability of the pool is the pool depth of only 50 documents and the fact that all runs are cross-lingual. Cross lingual runs usually have a worse retrieval performance, thus the pool will contain a lot of irrelevant documents.

We ran a similar test on the UPLIFT test collection in order to assess the sensitivity of performance measurements to the composition of the pool of the UPLIFT test collection. In theory the pool should be composed of very diverse systems, in practice the situation if often different. At TREC at least one run from each site is judged, and a lot of sites work with quite comparable systems. The UPLIFT pool also contains quite a bit of dependency. Therefore we created a new pool of a subset of eight systems, which differ substantially. We computed the average precision of each of these runs on this new pool and on a pool created on seven systems (leaving out the system of interest). We introduce a new reliability indicator: the *judged fraction*. The judged fraction (for the top n of a retrieval run) is defined as the percentage of documents of the top n which has been judged[19]. For the pool reliability experiment, we computed the judged fraction at rank 100, measured on the pool of the seven other systems. Table 4.9 shows the results

system	map(8)	map(7)	diff.	judged fr.@100
vAP1	0.3992	0.3964	0.0027	0.8779
vMA1	0.3861	0.3852	0.0010	0.9574
vSc1	0.3885	0.3874	0.0011	0.9609
vc1	0.2342	0.2282	0.0060	0.7497
vc4fow	0.3451	0.3425	0.0026	0.9426
vn	0.3101	0.3098	0.0003	0.8823
vp2	0.2493	0.2430	0.0063	0.7344
vsfow	0.2961	0.2915	0.0046	0.7888

Table 4.9. Pool validation experiment

of this experiment. We see that some runs (vc1, vp2 and vsfow) bring a lot of unique documents (27 documents of vp2's top 100 are unique) into the pool. However, having these documents judged does not really improve average precision. The maximum difference is 0.0063. We can draw two conclusions. First, most of the unique documents of vc1, vp2 and vsfow must have been judged non-relevant. More importantly, we see that considerable variations in judged fraction have a minor effect on the measurement error of the average precision.

[19]The judged fraction @ 100 for a judged run with pool depth=100 will always be 100%.

4.6. Conclusions

In this chapter we have motivated the evaluation methodology used for the experiments that will be discussed in the rest of this thesis. The focus is on the comparison of automatic IR systems in batch experiments, just like in the TREC evaluation conference. The advantage of such an approach is that the experiments can be more tightly controlled, the disadvantage is that it is difficult to extrapolate conclusions to settings where real users are involved. We have reviewed several performance measures and procedures for significance testing in order to define the evaluation procedure for the experiments in part II. We will summarize here the selected performance measures and validation procedures.

Performance measures. In this thesis we have chosen to work with the following measures:

Interpolated precision at fixed recall levels: These values are used to produce standard precision-recall graphs.

AP5-15: Averaged precision after 5, 10 and 15 documents. The measure gives a good insight in the *high precision*, the precision of the first screen of results. This is a measure which corresponds closely to the average user's perception of quality.

AVP: Average un-interpolated precision. This is the standard measure used in TREC, making it easier to make comparisons

R@R: recall at R. This measure is slightly more complicated than the recall at fixed cutoff level measure, but corrects for the often considerable variance in R, that makes the interpretation of cf. recall at 10 documents so difficult.

Statistical validation of results. Since there is a high variability of retrieval performance across topics, it is recommended to apply statistical significance tests. We tested the assumptions of several types of significance tests on data from IR experiments. Following Hull, we conclude that non-parametric tests can be applied for IR data, in particular the Friedman test for groups of runs and the sign test for a pair of runs. In the rest of this work, significance tests will be a standard part of our presentation. The tests will be based on sign tests and Friedman tests, with the Least Significance Difference as multiple comparison test. This test has the advantage of good power at a standard overall α level. We formulated several guidelines for the application of these tests, since it is easy to apply statistical tests and draw invalid conclusions. Another strategy for improving the reliability of inference is to do experiments with as much data points as possible, e.g. to compare effects across multiple test collections.

PART II

Applications

CHAPTER 5

Embedding translation resources in LM-based CLIR models

The application area of this chapter is Cross-Language Information Retrieval (CLIR). Cross-language retrieval is characterized by the fact that the query language is different from the language of the document collection. A CLIR system thus requires some translation component. There are several different types of translation resources available, which can be used as a basis for this translation step: machine translation, machine-readable translation dictionaries and parallel corpora. There are also different ways to combine these translation resources with a probabilistic IR model. We hypothesize that a CLIR model where translation and retrieval are tightly integrated will perform better than an approach where both are treated separately. In order to validate this hypothesis we propose several different probabilistic CLIR models, that each have a different degree of integration between translation and retrieval. These CLIR models have been evaluated on a test collection, crossed with several different instantiations of a translation resource. At the same time we investigated possible interactions between models and resources.

The chapter consists of three main parts: (i) an introduction providing the context for our main research hypothesis stating that integrated CLIR models can outperform CLIR methods where translation is carried out separately, (ii) a first series of experiments addressing the research questions derived from this hypothesis, and (iii) a second series of experiments addressing additional questions that were raised after analysing the first set of experiments. In some more detail: section 5.1 gives an overview of CLIR research, by discussing the CLIR task, different architectures and different translation resources. Section 5.2 discusses several ways to embed translation in generative probabilistic IR models. Section 5.3 describes the construction of simple word-based translation models using either Web-based parallel corpora or machine-readable dictionaries. Section 5.4 describes the main series of experiments which investigate the interaction between the different CLIR models and different translation resources. The results of these experiments gave rise to some additional research question, some of which are investigated in section 5.5. The main findings are summarized in section 5.6. Parts of this chapter have been published earlier in (Kraaij et al., 2003) and (Kraaij, 2003).

5.1. CLIR overview

Although CLIR had interested IR researchers already in the early seventies (Salton, 1973), it took another twenty years before the topic became a very active area of research. This increased interest is of course closely related to the fact that the global exchange of electronic information using the infrastructure of the World Wide Web became an everyday commodity. The proportion of people that have access to Internet is growing rapidly, especially in countries where English is not the mother tongue. This has the effect that the Web becomes more and more multilingual. It is difficult to estimate the non-English proportion of the Web, but conservative estimates state that non-English Web pages comprise about 30% of the Web and this percentage is growing. (Kilgariff & Grefenstette, 2003). It is thus obvious that there is a role for CLIR technology, although the role might be not as prominent as one would expect, since the majority of Internet-users seem to be satisfied with the available documents in their own language, the majority of Internet-users still being US citizens. A second important incentive for CLIR research has been the interest of the US intelligence community for the disclosure of Chinese and Arabic documents.

In this overview, we will define the part of CLIR functionality, which is the topic of our research (cross-lingual matching) and argue why a combination of MT and monolingual IR might not be the optimal way to tackle the problem (section 5.1.1), leading to a first informal discussion of the main research hypothesis of this chapter. Subsequently we discuss at some length the different options that have been investigated by other researchers to realize the desired functionality in terms of architectures (section 5.1.2) and translation resources (section 5.1.3). A more comprehensive discussion can be found in Oard & Dorr (1996). The overview concludes with a discussion of the challenges that a CLIR model has to face, since translation is not an easy task.

5.1.1. The role of translation in CLIR. A complete CLIR system requires translation functionality at several steps in order to help a user to access information, written in a foreign language. The first step is to relate the user's query to documents in the foreign language and to rank those documents based on a cross-lingual matching score. We will call this function *cross lingual matching*. The second step is to help the user to select documents from this ranked list. In a monolingual situation, this selection is usually done based on (query-oriented) document summaries. So a CLIR system could simply translate these summaries. The final step is the complete translation of the selected documents, which could be performed automatically or by a human translator, depending on the availability of MT for the language pair concerned and the required level of quality. Most CLIR research has been focused on the cross-lingual matching function although the recent CLEF conferences have initiated some work on the second function: (partial) translation for document selection (Oard & Gonzalo, 2002). The research reported in this chapter is also related to the cross-lingual matching function.

An easy method to implement CLIR is to use an MT system to translate the query from the query language into the document language and proceed with monolingual search. This functionality has been offered for Web search for a number of years. This solution can lead to adequate CLIR performance but has several disadvantages. The main disadvantage is that MT is only available for a selected number of language pairs. The

fact that automatic translations are not always well readable is not a big problem for cross-lingual matching, since documents and queries are usually reduced to a bag-of-words representation. A more important quality factor determining CLIR performance is whether each concept in the query is properly translated into the document language, which is not always the case for MT systems. An alternative approach is to use translation dictionaries, which are available for many language pairs. A good translation dictionary will list one or more translations for each word sense. Hull pointed out that one might be able to use synonym translations as a resource to improve retrieval performance (Hull, 1997). If the CLIR system is able to choose the correct word-sense and is able to deal with synonym translations in a robust way, it is quite likely that recall could be improved thanks to query expansion with synonyms. In this chapter we will compare different generative probabilistic models that can handle multiple translations in a robust way, by integrating translation more tightly in the retrieval model. Informally, our approach to the CLIR problem can be viewed informally as "cross-lingual (sense) matching". Both query and documents are modelled as a distribution over semantic concepts, which in reality is approximated by a distribution over words. The challenge for CLIR is to measure to what extent these distributions are related. The distributions are estimated on the available data. Since the amount of text to estimate distributions is very small, we think that dictionary-based CLIR methods have an advantage over MT-based methods, because they may help to produce smoothed distributions. This can either result in a better precision (in case the MT translation is wrong) or in higher recall, since synonym translations help to retrieve documents using related terminology.

Based on the hypothesis that accommodating multiple translations can yield CLIR systems with a higher effectiveness than systems that choose just one translation, we have formulated several research questions. The main questions are: (i) How do CLIR systems based on (word-by-word) translation models perform w.r.t. reference systems (e.g. monolingual, or MT-based query translation)? (ii) Which manner of embedding a translation model is most effective for CLIR? Before discussing the models (section 5.2), the complete set of research questions (section 5.4.1) and experiments (section 5.4), we will first continue with an overview of the different approaches to CLIR on the basis of a structural (section 5.1.2) and a resource-based classification (section 5.1.3) along the lines of Oard (1997). The structural classification makes a distinction between CLIR methods based on what textual elements (queries or documents) are translated, the resource-based classification is based on the type of resource which is used for the translation step e.g. an MT system, machine readable dictionaries or parallel corpora. In fact, this is a simplification of a quite extensive panorama of possible CLIR systems. In this chapter, we will do some comparison of resources, but mostly concentrate on the comparison of CLIR models based on a single resource.

5.1.2. Translating the query, documents or both. The following main approaches to CLIR can be distinguished:

Query translation. The most straightforward and most popular approach to the CLIR matching problem is to translate the query using a bilingual dictionary or MT system. Because we are only considering automatic approaches, the bilingual dictionary should be available in electronic form. The advantage of this approach is that only the query

has to be translated, so the amount of text that has to be translated is small. A disadvantage is that disambiguation of a very short query and selecting the proper translation is difficult, because the context is limited. However, it may not be necessary to do explicit sense disambiguation for an effective cross-language matching task. Query translation, can be improved by consulting the user during translation, an option that is clearly not available for document translation.

Document translation. Theoretically, it seems that document translation would be superior to query translation. Documents provide more context for resolving ambiguities and the translation of source documents into all the query languages supported by the IR system effectively reduces cross language retrieval to a monolingual task. Furthermore, document translation has the added advantage that document content is accessible to users in different languages (one of which may even be their mother tongue). Document translation, is inherently slower than query translation but, unlike query translation, it can be done off-line and translation speed may therefore not be crucial. Document translations need to be stored for indexing, though, and storage space may be a limiting factor, especially if many languages are involved. For realistically sized CLIR document collections (e.g. TREC 2GB), document translation is usually not considered a viable option, the majority of CLIR systems therefore apply a form of query translation, cf. (Voorhees & Harman, 2000a). Nevertheless, several studies have shown the great potential of document translation: with a fast statistical MT system optimised for CLIR (output is not legible) (Franz et al., 1999; McCarley & Roukos, 1998), or with massive use of commercial MT systems (Oard, 1998; Braschler & Schäuble, 2001). These studies show that applying MT to documents instead of queries results in considerably higher effectiveness. The study in McCarley & Roukos (1998) is especially convincing, since it is based on a comparison of two statistical translation models, trained on word-aligned sentence pairs. The first model is a variant of Model 3 of Brown et al. (1993) and is based on two conditional probabilities: the fertility[1] and the translation probability. Second key factor is that both conditional probabilities depend on a very small context (left and right neighbour term of the source term. This model performs convincingly better (19% improvement) than a simpler model, which does not use context and has a unary fertility for all source terms. An implementation variant of document translation is *index translation*, here only the content descriptors (which could be multi-word terms, like in the Twenty-One system (ter Stal et al., 1998)) of the document are translated. This strategy has the advantage that possibly less advanced translation tools are necessary (since only content descriptors are translated), but that the full document context is still available at translation time. The content descriptors could be used for document selection as well. However, an IR system based on multi-word index terms (e.g. noun phrases) does not scale very well.

Combining Query & document translation. A common technique in automatic classification tasks is to combine representations or methods to improve performance. Several groups have done experiments with a combination of query translation and document translation, usually with good results (Franz et al., 2000; Braschler & Schäuble, 2001).

[1]Fertility is defined as the number of words in the translation that constitute its translation. E.g. the fertility of 'not' in an English to French translation setting is two ('ne pas').

These experiments are based on data fusion: the RSV of each document is a combination of a query translation and a document translation run. Several factors probably play a role here: as we already argumented for document translation, the full context of a document can help with sense disambiguation. An even more important aspect might be the fact that different translation resources are combined. We will see that lexical coverage plays a decisive role for CLIR so a combination of resources has the potential of compensating for omissions in the individual translation resources.

Translation to an interlingua. A theoretically very attractive approach is to translate both queries and documents into an interlingua. An interlingua is a language independent representation, which is semantically unambiguous. An interlingua can have different levels of sophistication, depending on its use in an application, ranging from a logical language to a language independent concept hierarchy, e.g. EuroWordnet. In the IR framework, this approach is sometimes referred to as conceptual indexing, since the indexing language consists of artificial unambiguous concepts. Although this seems an attractive option, since queries and/or documents only need to be translated once and only one index needs to be maintained, in practice this last option is hardly ever used in other than very small scale, semi-automatic systems for well-defined domains, e.g. Ruiz et al. (2000), because devising and maintaining such an interlingua for applications with very diverse documents, e.g. WWW search engines, is not feasible. Also, translation would require a disambiguation step, while large scale disambiguation tools are not available yet.

Transitive translation. In certain cases, it might be useful or even necessary to use an intermediate or pivot language in the translation process, for example when direct transfer dictionaries do not exist. E.g. the only feasible option to translate Finnish queries or documents into Korean, might be based on English as a pivot language. Hiemstra, Kraaij & Pohlmann used Dutch successfully as a pivot language for query translation in a series of CLIR evaluations (TREC6-8,CLEF200,CLEF2001). Indirect word-by-word translation via a pivot language bears the possibility that a translation in the target language is found via different intermediate translations in the pivot language. One could interpret this as a reinforcement of a translation and consequently give it a higher weight, cf. Hiemstra & Kraaij (1999); Kraaij et al. (2000) and section 5.3.2. The extra translation step introduces additional ambiguity, so transitive translation will in general be less effective than direct translation. However, most of the performance loss can be compensated by using combinations of different transitive translation paths, e.g combining EN-FR-DE and EN-IT-DE. (Gollins & Sanderson, 2001).

No translation. At first sight, this approach seems to be a joke. Surprisingly, the approach performs much better than random retrieval for related languages, like English and French. Buckley realized that English words are often from French origin, some words still have identical forms (*promotion*) others stem from the same root (*emotion/émotion, langue/language*). The meaning of these pure and near *cognates* is often related. The close morphological distance can be exploited by treating English as misspelled French: each English query term is expanded with morphologically related French terms (Buckley et al., 1998). The expansion terms were allowed to have a small edit-distance and two equivalence classes were defined: one for vowels {a-e-i-o-u}and one

for k-sounds {c-k-qu}. The cognates-based bilingual English to French system yielded 60% of the monolingual French run on the TREC6 test collection.

5.1.3. Translation resources. There are different types of resources to implement the translation step of a CLIR system, but not all are available evenly across language pairs. As we have discussed earlier, an easy way out is to use a MT system to translate the query and use the result as input for a monolingual system. Apart from the fact that MT is not available for all language pairs and systems are costly, there are other reasons to consider dictionaries and parallel corpora as alternative or additional translation resources. Since dictionaries give multiple translations, a translated query could potentially have a query expansion effect, improving upon an unexpanded translation. Furthermore, MT systems contain a lot of machinery to produce morphologically and grammatically correct translations. This is totally useless for current state of the art IR systems, since they operate on a bag of words representation and often will apply some form of morphological normalisation. For these reasons, an MT system might be less robust than a simple word-by-word translation step based on a dictionary. From a CLIR point of view, the transfer dictionaries of an MT system are an interesting resource, though they usually cannot be accessed directly.

Dictionary-based approaches. A popular approach to CLIR is the use of a transfer dictionary, usually for query translation. There are many dictionaries available in electronic form. There are free dictionaries available on the web, with varying quality. Since the production of paper dictionaries is more and more based on electronic repositories, high quality translation dictionaries now become available via organisations like ELRA and LDC, albeit that quality has its price. The difficulty here is of course that dictionaries in electronic form are even more susceptible to piracy than software in object code, since the lexical knowledge can be encapsulated in derived products. Machine readable dictionaries provide translations for lemmas only, so a CLIR approach based on a transfer dictionary also requires a component for morphological normalisation in both the source and target language. This problem is related but not exactly equivalent to the techniques we discussed in chapter 6 about Conflation. Conflation is about defining equivalence classes for morphologically related words. The problem of dictionary-based CLIR is essentially a lexical lookup problem. We have to find the proper lemma in the dictionary-based on a wordform. Apart from all the problems with lexical lookup we already discussed in relation to MT, we want to add the problem of mismatch between transfer dictionaries and morphology. The mismatch can manifest itself at different levels e.g. a lemmatizer based on American English in combination with a transfer dictionary based on British English or differences in lexical coverage between the morphological component and the transfer dictionaries. Several groups have shown that reasonably effective CLIR is still possible when full morphological analysis is not available. One solution is to use n-grams as indexing terms (McNamee & Mayfield, 2001), which is especially popular to overcome the problem of the lack of a proper module for compound analysis(Savoy, 2002). Another option is to learn morphological rules (stem+suffix) from the data set itself using unsupervised methods like Minimum Description Length or rule induction (Goldsmith, 2001). Finally, if stemmers are available, one could stem both the query and the dictionary entries (de Vries, 2001). This is of course not optimal since it exhibits

under- and over-stemming problems. Still it is a simple and reasonably effective method when a full morphological analysis component is not available.

An early influential study on dictionary-based query translation was done by Hull & Grefenstette (1996). The conclusion of the study - based on a detailed error analysis - is that there is one main factor causing reduced retrieval effectiveness of a CLIR system in comparison with a monolingual system: "the correct identification and translation of multi-word terminology". Ambiguity is a much smaller but still significant factor, according to this study. We think that correct translation is important for all content terms. In our experience, one of the main determinants of CLIR system performance is lexical coverage.

A recent study on dictionary-based query translation (Diekema, 2003) categorizes the different problems of word-by-word translation in a taxonomy. A large number of queries were coded according to the taxonomy and a multiple regression test was carried out to quantify the effect of the various translation 'events' on retrieval performance. Several classes were shown to have a significant impact on retrieval performance, although the impact was small in comparison with query variability. We think that a more careful integration of word-by-word translation with the IR model (i.e. by normalizing termweights or using one of the models presented in section 5.2) could improve results.

Corpus-based approaches. Unlike dictionaries, which provide direct access to lexical transfer relations, *parallel corpora* provide indirect access to translation relations. The most famous example of a parallel text is of course Rosetta's stone, which helped to decipher (=translate) hieroglyphs. Research in the statistical MT tradition has shown that probabilistic transfer dictionaries can be derived from a sentence aligned corpus (Brown et al., 1990; Simard et al., 1992; Dagan et al., 1993; Hiemstra et al., 1997). The simplest models assume a one-to-one mapping between words. Iterative optimisation algorithms like the Expectation Maximisation (EM) algorithm usually form the basis. More complex algorithms have been devised to derive m to n translation relations (Brown et al., 1993). Corpus-based approaches to NLP only work well when there is enough training data. The difficult cases are usually rare cases, so (like in language modelling for speech recognition) huge amounts of parallel text are required to infer translation relations for complex lemmas like idioms or domain specific terminology. The former are probably not so important for CLIR since idioms are rarely used in queries. A more important drawback but also advantage of the corpus-based approach is the fact that corpora are usually domain dependent. Probabilistic dictionaries derived from parallel corpora will thus cover a smaller domain than general purpose dictionaries, but can potentially have a more thorough coverage of a particular domain. There is not always a good match between the domain of the parallel corpus and the domain of the target document collection. However, both types of dictionaries (corpus-based and human produced) could play a complementary role (Hull & Grefenstette, 1996).

Strictly parallel corpora are not always available, therefore special alignment algorithms have been designed to work with noisy aligned text or non-parallel but *comparable corpora*. Comparable corpora are document collections in different languages, which do not correspond one-to-one but cover the same domain or time period. These collections can be used to augment existing bilingual dictionaries. The idea is that in the same domain, words have comparable contexts in both languages. This fact can be exploited

for automatic dictionary construction by an algorithm, which compares the contexts of unknown words (Fung, 2000). Comparable corpora have also been used directly for CLIR, in the sense that they served directly as a resource for cross-lingual matching, without extracting a translation dictionary first. For certain multilingual document collections (e.g. news data), it is possible to align documents in different languages, since the documents have "the same communicative function" (Laffling, 1992), e.g. discuss the same event or news topic. Such a collection is called a *corpus of comparable documents*.

This process of document alignment has been carried out on a collection of news documents from the Swiss Press Agency (SDA), available in German, French and Italian (Braschler & Schäuble, 2000, 2001). A considerable portion of this collection was successfully aligned at the document level, using an algorithm based on dates and manually assigned language-independent content descriptors. The documents themselves, however, are not parallel, just comparable. The resulting collection could be called a parallel collection of comparable documents. Such a collection can be used to derive word associations across the language barrier, which in turn can be used for cross-language retrieval. These word associations, which form a "similarity thesaurus", can be computed by indexing the vocabulary of the multilingual collection by the id's of the aligned documents. Query translation can then be performed by finding the most similar terms in the similarity thesaurus (and possibly filtering out noisy terms using a target language dictionary). Experiments with the TREC and CLEF collections have shown that this is a viable approach, although not really competitive with MT or dictionary-based approaches (Sheridan & Ballerini, 1996; Braschler & Schäuble, 2000, 2001). The main disadvantage of the approach is that it is difficult to acquire a comparable corpus which subsumes the domain / time period of the query collection. However, this argument holds even stronger for real parallel corpora. The lexical coverage aspect (i.e. the main problem of CLIR according to Grefenstette) seems to be primordial for all approaches to CLIR. Most recent experiments with similarity thesauri indicate that these resources can sometimes help to improve the performance of an MT-based CLIR run (Braschler et al., 2002). Most probably, by filling in some lexical gaps of the MT lexicon.

In section 5.1.4, we stated that lexical coverage is one of the most important determinants of CLIR effectiveness. Several researchers have shown that effectiveness can be improved by using a combination of translation resources. However, most of these studies do not systematically study the interaction of the quality of resources with retrieval effectiveness or the interaction of resource types and CLIR models. An exception is the recent study of McNamee & Mayfield (2002) about the influence of lexical coverage. We will study the latter research question (interaction between resource types and their possible embeddings in a CLIR model) in some more detail in section 5.4.

The most recent work using comparable corpora for CLIR is from Lavrenko et al. (2002b). He recast the old idea of using a similarity thesaurus for CLIR (Sheridan & Ballerini, 1996) in a language model framework. The central idea is to estimate a so-called relevance model, i.e. a probability distribution over terms, supposing takings samples from relevant documents. This distribution can be estimated by exploiting co-occurrence information in documents, i.e. estimating the joint distribution of a term and the query. Lavrenko shows that relevance models can be estimated successfully in a different language by using comparable documents or a translation dictionary.

5.1.4. Challenges for CLIR systems. Not all of the problems well known from the field of MT are relevant for translation for CLIR, since the translation functionality for cross-lingual matching is a task secondary to the matching function itself. Matching is usually based on bag-of-words representations, so problems like lexical and structural differences between languages can be ignored for CLIR. Lexical ambiguity on the other hand is a central problem for CLIR and even to some extent for monolingual IR (cf. Section 3.3.3). A possible but difficult solution would be to index both queries and documents by an unambiguous sense representation. Another solution is to construct a sense preserving translation component, which is based on some disambiguation component. A third option is to omit disambiguation and exploit IR techniques (structured queries and/or term co-occurrence) to bypass ambiguity. This is the approach originally proposed by Hull and reformulated in a probabilistic framework by Hiemstra that we will follow in the rest of this chapter.

Structured queries have been proposed by several researchers as a solution for the problem of multiple translations for dictionary-based query translation. Hull (1997) proposed a quasi Boolean structure and Pirkola proposed to treat translation alternatives as synonyms (Pirkola, 1998; Pirkola et al., 1999). The idea behind these approaches is that naive dictionary-based translation is sub-optimal, since terms with many translations dominate the translated query. There seems to be an inverse correlation between term importance and its number of translation alternatives. Indeed a highly specific term usually has just one translation. Hull proposed to group translation alternatives together and to exploit the target corpus for disambiguation by favouring documents that contain at least one translation alternative from each query facet (concept). This might be a good approach for shorter queries, it is certainly not optimal for longer queries; recall will decrease substantially since there are many relevant documents that do not contain an instance of all facets. This effect can be overcome by using the (unspecified) quasi-Boolean approach or INQUERY's SUM operator. In the LM-based approach, the level of coordination between facets can be controlled by the smoothing parameter. In the absence of smoothing, the LM-based IR model realizes full coordination between query concepts. The more smoothing is applied, the fewer coordination is induced (see Hiemstra, 2001). The result of improved effectiveness demonstrated in Pirkola (1998) is difficult to generalize, since the study is based on a small test collection of only 34 topics in the medical domain. The concepts in these queries exhibit little polysemy since they are mostly highly domain specific.

The second important problem for CLIR is the translation of multi-word units (Hull & Grefenstette, 1996). E.g. *space probe/sonde spatiale* This problem is even more prominent when one of the languages is a compounding language (e.g. German or Dutch) and the other language not. It is very important that a CLIR system recognizes multi-word-units and treats these units as a whole, since otherwise important concepts are mistranslated. Multi-word-units are often important query terms, because of their specificity.

Our research will be mostly ignoring the problem of multi-word-units because its focus is an investigation of the properties of some simple CLIR models based on generative language models, using word-by-word translation. As already stated, we are seeking

to leverage the potential of synonym translations, while neutralizing the effect of incorrect translations. Starting point of our investigation will be the LM-based CLIR model, originally proposed by Hiemstra (Hiemstra, 2001). This model uses a simple dictionary-based word-by-word translation function, which is an integral part of the CLIR model. It is very well possible though, to combine more complex translation models with IR that can cope with the translation of multi-word-units to some extent. An example of context-sensitive query translation is Federico & Bertoldi (2002). Federico's CLIR system produces the e.g. 1, 5 or 10 best translations of a query using a bigram language model of the document language. Indeed using five translations instead of one does increase retrieval effectiveness for long queries (using title, description and narrative) but at the expense of significantly decreased efficiency. An example of context sensitive document translation involving complex translations is Franz et al. (1999). The IBM group built a fast MT system, with relaxed constraints with respect to correct word-order. The MT system is based on both a fertility and a contextual sense model (4-grams). This approach separates CLIR in a translation and a matching step, in order to reduce complexity. The net effect of this operation is that the translation step is optimized, but that retrieval performance might not be optimal, since synonym translations are not really exploited.

Since we will restrict our experiments to word-by-word translation in order to concentrate our research it is interesting to take the three main problems of dictionary-based CLIR as identified by Grefenstette as reference points for our research questions (Grefenstette, 1998). He formulated these main problems as follows:

1. *"Finding translations":* The most important property for a translation resource in the context of CLIR is lexical coverage of both the source and target language. If a term cannot be translated, this will almost definitely deteriorate retrieval effectiveness. In addition to the problem of acquiring translation dictionaries with sufficient coverage, a dictionary-based approach will face the same problems that MT faces: the translation of collocations, idiom and domain specific terminology. These classes require a more sophisticated morphological analysis and especially the domain specific terms challenge the lexical coverage of general purpose bilingual dictionaries. A second important class of terms, which can pose problems for lexical lookup is the class of proper names. Named entities like names of persons or locations are frequently used in IR queries and their translation is not always trivial. Often, the more commonly used geographical names like countries or capitals have a different spelling in other languages (Milan / Milano / Milaan), or translations that are not even related to the same morphological root (Germany / Allemagne / Duitsland). The names of organisations and their abbreviations are also a notorious problem, e.g. the United Nations can be referred to as UN , ONU, VN etc. (disregarding the problem of the morphological normalisation of abbreviations). When names have to be translated from languages in a different script like Cyrillic, Arabic or Chinese, this problem is even more acute. The process to define the spelling of a word in a language with a different script is called *transliteration* and is based on a phonemic representation of the named entity. Unfortunately, different "standards" are used for transliteration, e.g. the former Russian president's name in Latin script has been transliterated as Jeltsin, Eltsine, Yeltsin, Jelzin

etc. The reverse process: translating a transliterated term back to its original is even more difficult, since transliteration itself is ill-defined. However, automatic approaches based on Bayesian models trained on bilingual transliteration lists can outperform human translators (Knight & Graehl, 1997).

2. *"Pruning translation alternatives":* A word often has multiple translations, due to either sense ambiguity (two are more concepts are represented by homographs in the source language) or (near) synonymy in the target language. Translations based on word senses which are inappropriate for the context should be discarded. However, expanding the translated query with translations for closely related word-senses and synonym translations in the target language will probably help to improve recall, so we would like to keep those. Our hypothesis is that it is more important to keep the good translations than to discard the bad translations, since it is hard to recover from a missing good translation. Therefore it seems wise to start with all translations and remove translations in a conservative fashion. We will investigate this process in more detail in section 5.4.
3. *"Weighting translation alternatives":* Closely related to the previous point is the question of how to relate the translation alternatives. Term weighting is of crucial importance in IR. CLIR is not different in that respect, especially since we sometimes use quantitative estimates of the probability of a certain translation. Pruning translations can be seen as an extreme Boolean way of weighting translations. The intuition is that, just like in query expansion, it might be beneficial to assign a higher weight to the "main" translation and a lower weight to related translations. It is attractive to capture these intuitions about weights in a probabilistic framework, although it is not always straightforward how to estimate the translation probabilities. We will elaborate on this aspect in section 5.2.

After this discussion of different approaches to CLIR and the challenges that CLIR models have to deal with, we will proceed with a description of several alternative simple models for CLIR based on word-by-word translation. We will show that a proper probabilistic embedding of multiple translation alternatives into the retrieval model can indeed improve retrieval effectiveness. We will study different models using machine readable dictionaries and parallel corpora mined from the web and investigate the relative importance of finding, pruning and weighting translations.

5.2. Embedding translation into the IR model

In this section we will describe several ways to integrate word-by-word translation in a generative probabilistic retrieval model. Starting point of this work is Hiemstra's CLIR model. But the intuitions behind the variant models that we will describe (based on cross-entropy) and their formalization are slightly different. This section provides the theoretical background that we need for the experiments that are described in section 5.4.

When CLIR is considered simply as a combination of separate MT and IR components, the embedding of the two functions is not a problem. However, as we explained

in section 5.1.1, there are theoretical motivations for embedding translation into the retrieval model: since translation models usually provide more than one translation, we will try to exploit this extra information, in order to enhance retrieval effectiveness. This approach poses extra demands on the IR model, since it is well known that simple substitution of query terms by their translation results in poor performance. In section 2.6.3.5 we described that a monolingual probabilistic IR model based on a normalized log-likelihood ratio can be interpreted as measuring the cross-entropy between a unigram language model for the query and a model for the document, normalized by the cross-entropy between the query and collection model. We will repeat the cross-entropy reduction ranking formula here:

$$CER(Q;C,D) = H(Q,C) - H(Q,D) = \sum_{i=1}^{n} P(\tau_i|Q) \log \frac{P(\tau_i|D_k)}{P(\tau_i|C)} \tag{63}$$

where $P(\tau_i|Q)$ is the unigram language model estimated for the query (representing the user's view of relevant documents), $P(\tau_i|D_k)$ is the language model representing the document and $P(\tau_i|C)$ models the background language.

In the following subsections, we will describe several ways to extend this monolingual IR model with translation. Before measuring the cross entropy between query and document language models, both models have to be expressed in the same language. This can be achieved by either "translating" (or mapping) the query language model from the query language into the document language before measuring the cross-entropy, or by a "translation" of the document model from the document language into the query language. Since the MT literature speaks usually of source and target language and uses the symbols s and t for for words or sentences in source and target language, we have chosen to work with these symbols and terminology as well. So when we speak of source language, this will always refer to the query language and target language will always refer to the document language. This could be confusing when the translation direction is from the document language into the query language (from target into source).

The headers of the following sections (describing different CLIR models) contain run tags in parentheses, that will be used in section 5.4 to describe the experimental results. We will omit the normalization with the background model in the rest of the discussion, since it is a constant and does not influence document ranking for the different models.

5.2.1. Estimating the query model in the target language (QT). Instead of translating a query before estimating a query model (e.g. by using an MT system), we propose to directly estimate the query model in the document language. This can be achieved by decomposing the problem into two components that are easier to estimate:

$$P(t_i|Q_s) = \sum_{j}^{S} P(s_j, t_i|Q_s) = \sum_{j}^{S} P(t_i|s_j, Q_s) P(s_j|Q_s) \approx \sum_{j}^{S} P(t_i|s_j) P(s_j|Q_s) \tag{64}$$

where S is the size of the source vocabulary. Thus, $P(t_i|Q_s)$ can be approximated by combining the translation model $P(t_i|s_j)$, which we can estimate e.g. on a parallel corpus, and the familiar language model $P(s_j|Q_s)$ which can be estimated using relative frequencies.

This simplified model, from which we have dropped the dependency of $P(t_i|s_j)$ on Q, can be interpreted as a way of mapping the probability distribution function in the

source language event space $P(s_j|Q_s)$ onto the event space of the target language vocabulary. Since this probabilistic mapping function involves a summation over all possible translations, mapping the query model from the source language can be implemented as the matrix product of a vector representing the query probability distribution over source language terms with the translation matrix $P(t_i|s_j)$. The result is a probability distribution function over the target language vocabulary.

Now we can substitute the query model $P(\tau_i|Q)$ in formula (46) with the target language query model in (64) and, after a similar substitution operation for $P(\tau_i|C)$, we arrive at CLIR-model QT (Query "Translation"):

$$\text{(65)} \quad \text{QT:} \quad CER(Q_s; C_t, D_t) = \sum_{i=1}^{n} \sum_{j=1}^{S} P(t_i|s_j) P(s_j|Q_s) \log \frac{(1-\lambda)P(t_i|D_t) + \lambda P(t_i|C_t)}{P(t_i|C_t)}$$

5.2.2. Estimating the document model in the source language (DT). Another way to embed translation into the IR model is to estimate the document model in the query (source) language:

$$\text{(66)} \qquad P(s_i|D_t) = \sum_{j}^{T} P(s_i, t_j|D_t) = \sum_{j}^{T} P(s_i|t_j, D_t) P(t_j|D_t) \approx \sum_{j}^{T} P(s_i|t_j) P(t_j|D_t)$$

where T is the size of the target vocabulary. Obviously, we need a translation model in the reverse direction for this approach. Now we can substitute (66) for $P(\tau_i|D)$ in formula (46):

$$\text{(67)} \quad \text{DT:} \quad H(Q_s; C_t, D_t) = \sum_{i=1}^{n} P(s_i|Q_s) \log \frac{\sum_{j=1}^{T} P(s_i|t_j)((1-\lambda)P(t_j|D_t) + \lambda P(t_j|C_t))}{\sum_{j=1}^{T} P(s_i|t_j) P(t_j|C_t)}$$

So, though this model has been often described as a model for query translation (e.g. Hiemstra (2001)), we would rather view it as a CLIR model based on a simple form of document translation (using a word-by-word approach), which on the basis of document terms generates a query. However, contrary to other document translation approaches like Oard (1998) and Franz et al. (1999), only those terms in the document are translated that do lead to a match with query terms. It is therefore a more efficient and more scalable approach.

It is important to realize that both the QT and DT models are based on context insensitive translation, since translation is added to the IR model after the independence assumption (32) has been made. Recently, a more complex CLIR model based on relaxed assumptions - context sensitive translation but term-independence based IR - has been proposed in Federico & Bertoldi (2002). In experiments on the CLEF test collections, the aforementioned model also proved to be more effective than a probabilistic CLIR model based on word-by-word translation. However, it has the disadvantage of reducing efficiency due to a Viterbi search procedure.

The idea of embedding a translation step into an IR model based on query likelihood was developed independently by several researchers (Hiemstra & de Jong, 1999; Kraaij et al., 2000; Berger & Lafferty, 2000). Initially, translation probabilities were estimated from machine-readable dictionaries, using simple heuristics (Hiemstra et al., 2001a). Other researchers have successfully used models similar to DT, in combination

with translation models trained on parallel corpora, though not from the Web (McNamee & Mayfield, 2001; Xu et al., 2001).

5.2.3. Overview of variant models and baselines. In this subsection we will discuss several variant instantiations of QT and DT, which help us measure the importance of the number of translations (pruning) and the weighting of translation alternatives. We also present several baseline CLIR algorithms taken from the literature and discuss their relationship to the QT and DT models.

External translation (MT, NAIVE). As we already argued in the section 5.1.1, the simplest solution to CLIR is to use an MT system to translate the query and use the translation as the basis for a monolingual search operation in the target language. This solution does not require any modification to the standard IR model as presented in formula (63). We will refer to this model as the *external* (query) translation approach. The translated query is used to estimate a probability distribution for the query in the target language. Thus, the order of operations is: (i) translate the query using an external tool; (ii) estimate the parameters $P(t_i|Q_t)$ of a language model based on this translated query.

In our experimental section below, we will list results with two different instantiations of the external translation approach: (i) MT: query translation by Systran, which employs a high-level linguistic analysis, context-sensitive translation (i.e. disambiguation), extensive dictionaries etc. (ii) NAIVE: naive replacement of each query term by its translations (not weighted). The latter approach is often implemented using bilingual word lists for CLIR. It is clear that this approach can be problematic for terms with many translations, since they would then get a higher relative importance. The NAIVE method is only included here as a baseline for the weighted models and helps to study the effect of the number of translations on the effectiveness of various models.

Most probable translation (QT-MP). There are different possible strategies to prune the translation alternatives that are given by the translation model. An extreme pruning method is to keep just the most probable translation. (cf. section 5.3.1.2 for other pruning strategies). A translation model for query model translation based on taking the most probable translation of each query term (QT-MP) could also be viewed as an instance of the external translation model, but one that uses a corpus-based disambiguation method. Each query term is translated by the most frequent translation in a parallel corpus, disregarding the query context.

Equal probabilities (QT-EQ). If we don't know the precise probability of each translation alternative for a given term, the best thing to do is to fall back on uniform translation probabilities. This situation arises, for example, if one works with bilingual dictionaries. We hypothesize that this approach will be more effective than NAIVE, since translation probabilities are properly normalized, but less effective than QT since each translation has the same weight.

Synonym-based translation (SYN). An alternative way to embed translation into the retrieval model is to view translation alternatives as synonyms. This is partly true. For lemmas that are not ambiguous, translation alternatives are indeed (near) synonyms. However, in the case of polysemy, alternative translations have a different meaning and are clearly not synonymous. Strictly speaking, when terms are pure synonyms, they can be substituted in every context. Combining translation alternatives with the synonym

operator of the INQUERY IR system (Broglio et al., 1995), which conflates terms on the fly, has been shown to be an effective way of improving the performance of dictionary-based CLIR systems (Pirkola, 1998). In our study of stemming algorithms (Kraaij & Pohlmann, 1996b), we independently implemented the synonym operator in our system. This online conflation function replaces the members of the equivalence class by a class id, usually a morphological root form. We have used this function to test the effectiveness of a synonymy-based CLIR model in a language model IR setting.

The synonym operator for CLIR can be formalized as the following class equivalence model (assuming that all translations t_j for term s_i are defined by the set $\sigma(s_i)$ and there are T unique terms in the target language):

$$P(\text{class}(s_i)|D_t) = \frac{\sum_{t_j} \in \sigma(t_j) c(t_j, D_t)}{\sum_j^T c(t_j, D_t)} = \sum_j^T \delta(s_i, t_j) P(t_j|D_t) \tag{68}$$

where $P(\text{class}(s_i)|D_t)$ is the probability that a member of the equivalence class of s_i is generated by the language model $P(\tau_i|D_t)$ and

$$\delta(s_i, t_j) = \begin{cases} 1 & \text{if} \quad t_j \in \sigma(s_i) \\ 0 & \text{if} \quad t_j \notin \sigma(s_i) \end{cases} \tag{69}$$

Here $c(t_j, D_t)$ is the term frequency (counts) of term t_j in document D_t.

The synonym class function $\delta(s_i, t_j)$ can be interpreted as a special instantiation of the translation model $P(s_i|t_j)$ in (66), namely $P(s_i|t_j) = 1$ for all translations t_j of s_i. Of course, this does not yield a valid probability function since the translation probabilities for all translations s_i of a certain t_j do not sum to one, because the pseudo-synonym classes are not disjunct due to sense ambiguity. But the point is that the structure of a probabilistic version of the SYN model is similar to the DT model, namely one where all translations have a reverse translation probability $P(s_i|t_j)$ equal to one. This is obviously just an approximation of reality. We therefore expect that this model will be less effective than the QT and DT models. In our implementation of the SYN model, we formed equivalence classes by looking up all translations of a source term s_i in the translation model $P(t_j|s_i)$. The translations receive weight 1 and are used as pseudo translation-probabilities in the model corresponding to formula (67).

5.3. Building the term translation resources

As said, the generation of well-formed target language expressions is not an issue in the context of CLIR. In our probabilistic framework translation can thus be performed on a word-by-word basis. As a consequence the role of translation resources is to translate between words. The translation model can thus be restricted to a matrix of translation probabilities between each word in the source language and each word in the target language, a probabilistic translation dictionary. In this section we will describe some procedures to generate these probabilistic dictionaries on the basis freely available resources. These will be compared with expensive high quality machine readable dictionaries.

5.3.1. Web-based translation models. Parallel corpora seem an ideal resource for the construction of translation models, since we can benefit from proven word alignment

techniques, which have been developed for statistical MT. Parallel texts are defined in the computation linguistics community as:

> texts accompanied by their translations in one or several other languages (Véronis, 2000)

Translation models can be derived after first aligning the sentences in source and target language text and subsequently aligning words using statistical algorithms that maximize a probabilistic criterion. Translation models can be derived easily from the word-aligned texts. A serious drawback of resorting to parallel texts as a translation resource is that it is difficult to acquire large parallel corpora for many language pairs. For many language pairs, large parallel corpora are not available, or access is restricted. This problem can partially be overcome by using the Web as a resource of parallel pages (Resnik, 1998; Nie et al., 1999). Many non-English Web sites offer English translations of their pages, which can form the basis for the construction of parallel corpora with English as one of the languages. Moreover, it is possible (with some degradation in quality) to combine translation models in order to translate between languages for which no parallel corpora (or even no dictionaries) exist.

The next two subsections describe the process of mining a probabilistic dictionary from the Web. The first step in this process is to find parallel texts on the Web.

5.3.1.1. *Mining parallel pages.* We have developed several parallel corpora based on parallel web pages for the CLEF 2001 evaluation in close cooperation with the RALI laboratory of the Université de Montréal. The PTMiner tool (Nie et al., 1999) was used to find web pages that have a high probability to be translations of each other. The mining process consists of the following steps:

Determining candidate sites: Query a Web search engine for Web pages with a hyperlink anchor text "English version" and respective variants.

Determine candidate page URLs: (For each web site) Query a Web search engine for all Web pages on a particular site.

Pair scanning: (For each web site) Try to find pairs of path names that match certain patterns, e.g.: `/department/tt/english/home.html` and `/department/tt/italian/home.html`.

Apply sanity check: (For each pair) download Web pages, perform a language check using a probabilistic language classifier, remove pages which are not positively identified as being written in a particular language.

The mining process was run for four language pairs and resulted in one large and three modestly sized parallel corpora. Table 5.1 lists sizes of the corpus during intermediate steps. It is striking that the number of candidate pairs is significantly reduced during the downloading and cleaning step. Due to the dynamic nature of the web, a lot of pages that have been indexed, do not exist anymore. Sometimes a site is down for maintenance. Finally, a lot of pages are simply place holders for images and are discarded by the language identification step. These parallel corpora have been used in different ways: (i) to refine the estimates of translation probabilities of a dictionary based translation system (Kraaij & Pohlmann, 2001) (ii) to construct simple statistical translation models (IBM model 1) Nie et al. (1999). In this chapter we will only report on the latter application.

language	# web sites	# candidate pages	# candidate pairs	# cleaned pairs
EN-IT	3651	1053649	23447	4768
EN-DE	3817	1828906	33577	5743
EN-NL	3004	1170082	24738	2907
EN-FR	n.a.	n.a.	n.a.	18807

Table 5.1. Intermediate sizes during corpus construction, n.a. = not available

5.3.1.2. *Building translation models.* Statistical machine translation is a data driven approach to translation. The central component of such an approach is a (statistical) translation model, which is trained on observed data and can subsequently be used to translate text. A series of models of increasing complexity have been developed at IBM (Brown et al., 1993), all based on the noisy channel paradigm (Shannon & Weaver, 1949). The core idea of applying the noisy channel model in linguistics is that a lot of problems can be cast as *decoding* problems, which is a central element from information theory. Instead of trying to determine the input on the basis of the output (e.g. determine the English translation of a French sentence) by directly estimating

(70) $$\hat{\mathbf{e}} = \underset{\mathbf{e}}{argmax}\ \ P(\mathbf{e}|\mathbf{f})$$

the noisy channel approach reverses the problem by applying Bayes' rule:

(71) $$\hat{\mathbf{e}} = \underset{\mathbf{e}}{argmax}\ \ P(\mathbf{e})P(\mathbf{f}|\mathbf{e})$$

Thus the problem can be decomposed in two subproblems: the estimation of $P(\mathbf{e})$ and $P(\mathbf{f}|\mathbf{e})$. The former is the source language model, which models the probability of sequences of input words. The latter is the channel model, which models the probability that the English sentence **e** could be at the origin of the observed sentence **f**. The advantage of the decomposition is that we can use $P(\mathbf{e})$ to model syntactical constraints and $P(\mathbf{f}|\mathbf{e})$ for the lexical aspects of translation.

Informally, the idea is that we try to guess the original information which was *transmitted* on the basis of the observed information and models of the noisy channel and of the source. There are multiple guesses possible, each with an associated probability resulting from a multiplication of the source model and channel model probabilities. Determining the most probable source information is called *decoding*.

For our experiments translation models were constructed on the basis of the parallel Web corpora that we described in the precious section. The construction of the translation models is documented in (Kraaij et al., 2003). Here, the major aspects will be summarized.

Format conversion: In this first step, the textual data is extracted from the Webpages. Of the HTML markup tags, only paragraph markers and sentence boundary information is retained, since these markers are important for the sentence alignment process.

Sentence alignment: After a pair of Web pages has been converted in neatly structured documents consisting of paragraphs consisting of sentences, the document pair is aligned. This alignment produces so-called *couples* i.e. minimal-size pairs of text segments from both documents. The couples usually consist

of two sentences, but sometimes a sentence cannot be aligned, or is aligned to more than one sentence. The alignment procedure we used was based on Simard et al. (1992)

Tokenization, Lemmatization and Stop words: Since the final goal of our procedure is a word-alignment, sentences have to be tokenized first. This is quite straightforward for Romance and Germanic languages using spaces and punctuation as word delimiters, but more complicated for languages like Chinese.

Since our goal is to use translation models in an IR context, it seems natural to have both the translation models and the IR system operate on the same type of data. The basic indexing units of our IR systems are word stems. Lemmatizing and removing stop words from the training material is also beneficial for statistical translation modeling, helping to reduce the problem of data sparseness in the training set.

Since we did not have access to full morphological analysis for Italian, we used a simple, freely-distributed stemmer from the Open Muscat project.[2] For French and English, we lemmatized each word-form by lookup in a morphological dictionary using its POS-label (assigned by a HMM-based POS-tagger (Foster, 1991)) as a constraint. As a final step, stop words were removed.

Word Alignment: Following common practice, only 1-1 aligned sentence pairs were used for the word alignment process. A simple statistical translation model: IBM's *Model 1* was trained on the pre-processed aligned sentences. This model disregards word order (which is ignored in most IR systems) and is relatively easy to train. As a by-product, the training procedure for Model 1 yields the conditional probability distribution $P(s|t)$, which we need for our CLIR model. The following table provides some statistics on the processed corpora.

	EN-FR	EN-IT
# 1-1 alignments	1018K	196K
# tokens	6.7M/7.1M	1.2M/1.3M
# unique stems	200K/173K	102K/87K
# unique stems (P > 0.1)	81K/73K	42K/39K

Table 5.2. Sentence-aligned corpora

Pruning the model: The $P(s|t)$ distribution is estimated on the corpus of aligned sentences, using the Expectation-Maximisation (EM) algorithm. As in any other corpus-based approach to learning properties of natural language data, sparseness poses a real problem. A complex model requires a large dataset in order to estimate parameters in a reliable way. IBM Model 1 is not a very complex model, but contains many parameters, since $P(s|t)$ covers the cross-product of source and target language vocabularies. Since the aligned corpora are not extremely large, translation parameters for which there is not much training data (rare English and French words) cannot be reliably estimated. This might not

[2]Currently distributed by OMSEEK: `http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/omseek/om/languages/`

be so dramatic as it sounds, since extremely rare words (like hapaxes) are less important for IR. We noticed from preliminary experiments, that the retrieval effectiveness of a CLIR system based on probabilistic model can be improved by deleting parameters (translation relations), for which indications exist that they are less reliable. From a machine learning viewpoint, this is not so surprising, since pruning is a well-known technique to increase robustness of a model. We have experimented with two common pruning methods:

Probability thresholding: Translation probabilities below a certain threshold are considered unreliable. Model parameters below an empirically determined threshold (0.1 yielded good results) are removed and remaining parameters are re-normalized. Although there is no direct correspondence to e.g. the marginal counts of the target or source word, this method works well.

Best N model parameters : Another possibility for pruning is to delete those parameters that contribute the least to the quality of the model. One way to measure quality is the normalized log-likelihood of a target language test corpus given a source language test corpus. The individual contribution of each parameter (translation probability) can be rated by computing the aforementioned log-likelihood based on the full translation model in comparison with the log-likelihood of the translation model where the parameter is set to zero. The log-likelihood ratio for a reliable parameter will be high, indicating that pruning such a parameter would seriously hurt the performance of the model (Foster, 2000). Pruning the model is than a matter of ordering, thresholding and re-normalizing.

The evaluation of the precision of the mining process has not been done in a systematic way for all language pairs. Inspection of the generated translation models revealed that the language identification process had not always worked effectively, since some target language terms were listed as a source language term in the generated dictionaries. A preliminary evaluation of the precision of the mining process (in terms of the proportion of correct pairs) is reported in (Kraaij et al., 2003).

Since translation models trained on parallel corpora will not have a complete coverage of names, we applied one back-off rule in the translation model: if a word is not found its translation is the identical form, in the hope that the target language translation is in fact a cognate. Fuzzy matching strategies might even improve recall.

5.3.2. Estimating translation probabilities for MRD's. Our dictionary-based query translation strategies are based on the Van Dale VLIS database. The VLIS database is a relational database which contains the lexical material that is used for publishing several bilingual translation dictionaries, i.e. Dutch → German, French, English, Spanish and Italian. The database is a richer resource than most bilingual term-lists, since it is used to produce bilingual dictionaries on paper. Not all of the information is relevant to our application, but we did use (among others) the part of speech information (to avoid selecting some wrong senses) and a style indicator, marking pejorative terms (to remove pejorative translations). The database contains 270k simple and composite lemmas for Dutch corresponding to about 513k concepts. The lexical entities are linked by several

typed semantical relations, e.g. hyperonymy, synonymy, antonymy, effectively forming a concept hierarchy. All concepts have one or more surface forms in Dutch and one or more translation alternatives in French, Spanish, German, English and Italian.

In table 5.3 below, some statistics for the VLIS database are given. We have prepared

Concepts	513k
Dutch	270k
English	265k
German	230k
French	248k
Spanish	147k
Italian	91k

Table 5.3. Number of translation relations in the VLIS database

several translation models based on the information in the VLIS database. All models are based on using just the simple lemmas. The basic idea is to look up all possible translations of a certain lemma. Both the search term and the translations[3] are normalized to minimize lookup problems, POS information for both search terms and lexical entries is available. Despite the morphological normalization and the availability of part-of-speech information, search terms are sometimes still not found, although (spelling) variants are listed in the dictionary. Therefore we included some back-off strategies to increase lookup effectiveness. The lookup strategy is roughly defined by the following steps:

(1) Lookup with syntactic restriction, if no translations found:
(2) Lookup without syntactic restriction, if no translation found:
(3) Lookup spelling alternatives: with/without initial capital, American/British English spelling variants etc. etc.. If no translations found:
(4) Leave unchanged

Each word sense has a main translation and some additional translations. The additional translations are often synonyms but can also be restricted to a particular context of the word sense. E.g. the Dutch verb “barsten” has two senses: the first sense has as main translation “crack” and an additional synonym translation “burst”, in the context of skin the best translation is “chap” . The second sense has as a main translation “burst” and as additional translation “explode”. Initially, we performed experiments with taking only the main translation, since we wanted to avoid less common (e.g archaic) translations. Before translation, topics are pre-processed in a series of steps in order to normalize them to a lemma format:

(1) Tokenizing: The query string is separated into individual words and punctuation characters.
(2) Part of speech tagging: Word forms are annotated with their part of speech. We use the Xelda toolkit developed by Xerox Research Centre in Grenoble for tagging and lemmatisation.

[3]We often used the translation relations in reverse direction.

(3) Lemmatizing: Inflected word forms are lemmatized (replaced with their base form).
(4) Stop word removal: Non-content bearing words like articles, auxiliaries etc, are removed.

The remaining query terms are subsequently translated into the various source languages. We used three different strategies for selecting translations from the VLIS database: all translations, the "most probable translation" without using context information and translation after disambiguation in the source language. The first two strategies will be discussed in the next sections, the disambiguation strategy has been presented in Kraaij & Pohlmann (2001). We also performed some experiments with different constraints for the selection of translations, which are reported in section 5.5.2.

More often than not a translation consists of more than one word. It can be a phrase or a list of alternatives, but also often some context is given in parentheses. A cleanup procedure has been defined based on a couple of heuristics: removing context in parentheses, removing punctuation and stop words, lemmatizing the remaining words, treating each as a separate translation. This procedure was used to make a clean version of the dictionary which was suitable for subsequent processing.

Due to polysemy, but also due to fine grained sense distinctions, which are important for translators, multiple senses are available for the majority of the lemmas, each again possibly with several translations. Since the VLIS lexical database does not contain any frequency information about translation relations, we can only approximate $P(t|s)$ in a crude way. Some lemmas have identical translations for different senses. The Dutch lemma *bank*, for example, translates to **bank** in English in five different senses: "institution", "building", "sand bank", "hard layer of earth" and "dark cloud formation". Other translations are **bench**, **couch**, **pew**, etc.

VLIS-query(English translations of *bank*(NL))

bank (institution), **bank** (building), **bank** (sand bank), **bank** (hard layer of earth), **bank** (dark cloud formation), **bench** (seat), **couch** (seat), **pew** (seat)

It is easy to compute the forward translation probability $P(t_j|s_i)$ for this (simplified) example: $P(\text{bench}|\text{bank}) = 1/8$. In a more formal way:

$$P(t_j|s_i) = \frac{c(s_i, t_j)}{\sum_j c(s_i, t_j)} \tag{72}$$

Here, $c(s_i, t_j)$ is the number of times the translation relation (s_i, t_j) is found in the lexical database.

The computation of the reverse translation probability $P(s_i|t_j)$ is slightly more elaborate. First, we select all lemmas in the target language that translate to the query term in the source language. We subsequently translate the target language lemmas to the source language and count the number of times that the target lemma translates to the literal query term, e.g.

VLIS-query(Dutch translations of English translation of *bank*(NL))

bank (English) → *bank (2x), oever, reserve, rij* etc.
pew (English) → *(kerk)bank, stoel*
couch (English) → *bank, sponde, (hazen)leger,* etc.

The probability that **bank** (E) translates to *bank* (NL) is twice as high as the probability that **bank** (E) translates to *oever*. The estimation of $P(s_i|t_j)$ on the VLIS database can be formalized as:

$$P(s_i|t_j) = \frac{c(s_i,t_j)}{\sum_i c(s_i,t_j)} \tag{73}$$

So far we have discussed translating from and two Dutch, which is the pivot language in the lexical database.

For transitive translation via Dutch as a pivot language (e.g. French to Italian), we investigated two estimation methods. The first estimation method disregards the fact that Dutch is used as a pivot language and is based on (72) and (73). The second estimation procedure explicitly models the individual translations steps, to and from the interlingua:

$$P(t_j|s_i) \approx \sum_k P(d_k|s_i)P(t_j|d_k) = \sum_k \frac{c(s_i,d_k)}{\sum_k c(s_i,d_k)} \frac{c(d_k,t_j)}{\sum_j c(d_k,t_j)} \tag{74}$$

$$P(s_i|t_j) \approx \sum_k P(d_k|t_j)P(s_i|d_k) = \sum_k \frac{c(d_k,t_j)}{\sum_k c(d_k,t_j)} \frac{c(s_i,d_k)}{\sum_i c(s_i,d_k)} \tag{75}$$

where d_k represents a term from the Dutch interlingua. We hypothesized that this more detailed estimation procedure would improve retrieval performance. We will give a symbolic example to show the difference between the direct and transitive estimation procedure. Suppose the French word $f1$ has two Dutch translations d_1 and d_2. Now d_1 has one English translation e_1 and e_2 has two English translations e_2 and e_3. The direct translation probability estimates for translating F1 into English are $P(e_1|f_1) = P(e_2|f_1) = P(e_3|f_1) = 1/3$. The transitive estimates are: $P(e_1|f_1) = \sum_i P(e_1|d_i)P(d_i|f_1) = 1/2$, and in a similar fashion: $P(e_1|f_2) = P(e_1|f_3) = 1/4$.

Surprisingly, the experiments with the simpler approach (direct estimation: (72) and (73)) yielded better results than (74) and (75), we therefore did not pursue the transitive probability estimates further. We hypothesize that the performance decrease is due to the fact that VLIS contains roughly twice as many concepts as lemmas. This means that in a transitive estimation procedure, the probability mass is spread equally over each sense. Now if some of the word senses are actually just sense variations (in other words, the sense differences are sometimes small and sometimes large), then the transitive estimation procedure will assign most probability mass to related word senses, which might down-weight clearcut word senses. The direct estimation procedure suffers less from this problem.

5.4. Experiments I

We carried out a series of contrastive experiments to gain more insight into the relative effectiveness of the various CLIR models presented in section 5.2.1 - 5.2.3 combined with translation models estimated according to the methods described in section 5.3. We will first outline our research questions in section 5.4.1, subsequently describe the experimental conditions, test collection and baseline systems in subsections 5.4.2- 5.4.4. Experimental results are presented in subsection 5.4.5 and discussed in relation to the research questions in subsection 5.4.6.

5.4.1. Research Questions. The main research hypothesis of this work is that using multiple translation alternatives can result in better CLIR performance than using just one translation if and only if translation is properly integrated into the retrieval model. This hypothesis will be studied by addressing the following research questions:

i): How do CLIR systems based on word-by-word translation models perform w.r.t. reference systems (e.g. monolingual, MT)?
ii): Which manner of embedding a translation model is most effective for CLIR? How does a probabilistically motivated embedding compare with a synonym-based embedding?
iii): Is there a query expansion effect and how can we exploit it?
iv): What is the relative importance of pruning versus weighting?
v): Which models are robust against noisy translations?
vi): Are there any differences between integrating a machine readable dictionary (MRD) or parallel web corpus as a translation resource in a CLIR system?

The first two questions concern the main goal of our experiments: What is the effectiveness of a probabilistic CLIR system in which translation models mined from the Web or estimated from a MRD are an integral part of the model, compared to CLIR models in which translation is merely an external component? The remaining questions help to understand the relative importance of various design choices in our approach, such as pruning, translation model direction etc.

5.4.2. Experimental conditions. We have defined a set of contrastive experiments in order to help us answer the above-mentioned research questions. These experiments seek to:

(1) Compare the effectiveness of approaches incorporating a translation model produced from the Web versus a monolingual baseline and an off-the-shelf external query translation approach based on Systran (MT).
(2) Compare the effectiveness of embedding query model translation (QT) and document model translation (DT).
(3) Compare the effectiveness of using a set of all-weighted translations (QT) versus just the most probable translation (QT-MP).
(4) Compare the effectiveness of weighted query model translation (QT) versus equally-weighted translations (QT-EQ) and non-weighted translations (NAIVE).
(5) Compare the effectiveness of treating translations as synonyms (SYN) with weighted translations (QT) and equally-weighted translations (QT-EQ).

(6) Compare different strategies for pruning translation models: best N parameters or thresholding probabilities.
(7) Run the model comparison experiments with translation models derived from the parallel Web corpora and the VLIS lexical database

Each strategy is represented by a run-tag, as shown in table 5.4. Table 5.5 illustrates the

run tag	short description	matching language	section
MONO	monolingual run	document	5.2, 5.4.4
MT	Systran external query translation	document	5.2.3, 5.4.4
NAIVE	equal probabilities	document	5.2.3
QT	translation of the query language model	document	5.2.1
DT	translation of the document language model	query	5.2.2
QT-MP	most probable translation	document	5.2.3
QT-EQ	equal probabilities	document	5.2.3
SYN	synonym run based on forward equal probabilities	query	5.2.3

Table 5.4. Explanation of the run tags

differences between the different translation methods. It lists, for several CLIR models, the French translations of the (English) word "drug". The translations in table 5.5 are provided by the translation models $P(e|f)$ and $P(f|e)$ estimated on the parallel Web corpus. Translation models can be pruned by discarding the translations with $P < 0.1$ and renormalizing the model (except for SYN) or by retaining the 100K best parameters of the translation model. The first pruning method (probability threshold) has a very different effect on the DT method in comparison with its effect on QT: the number of terms that translate into *drug* according to $P(e|f)$ is much larger than the translations of *drug* found in $P(f|e)$. There are several possible explanations for this: quite a few French terms, including the verb *droguer*, the compounds *pharmacorésistance, pharmacothérapie* etc., all translate into an English expression or compound involving the word *drug*. Since our translation model is quite simple, these compound-compound translations are not learned. [4] A second factor that might play a role is the greater verbosity of French texts compared to their English equivalent (cf. table 5.2). For the models which have been pruned using the 100K best parameters criterion, the differences between QT and DT are smaller. Both methods yield multiple translations, most of which seem related to *drug*; so there is a clear potential for improved recall due to the query expansion effect. Notice, however, that the expansion concerns both the medical and the narcotic senses of the word *drug*. We will see in the following section that the CLIR model is able to take advantage of this query expansion effect, even if the expansion set is noisy and not disambiguated.

[4] A more extreme case is query C044 about the "tour de france". According to the $P(e|f) > 0.1$ translation model, there are 902 French words that translate into the "English" word *de*. This is mostly due to French proper names, which are left untranslated in the English parallel text

run id	translation	translation model
MT	drogues	
QT	<drogue, 0.55; medicament, 0.45>	$P(f\|e) \leq 0.1$
QT-EQ	<drogue, 0.5; medicament, 0.5>	
QT-MP	<drogue, 1.0>	
SYN	<drogue, 1.0; medicament, 1.0>	
NAIVE	<drogue, 1.0; medicament, 1.0>	
DT	<antidrogue, 1.0; drogue, 1.0; droguer, 1.0; drug, 1.0; médicament, 0.79; drugs, 0.70; drogué, 0.61; narcotrafiquants, 0.57; relargage, 0.53; pharmacovigilance, 0.49; pharmacorésistance, 0.47 médicamenteux, 0.36; stéroïdiens, 0.35, stupéfiant, 0.34; assurance-médicaments, 0.33; surdose, 0.28; pharmacorésistants, 0.28; pharmacodépendance, 0.27 pharmacothérapie, 0.25; alcoolisme, 0.24; toxicomane, 0.23; bounce, 0.23; anticancéreux, 0.22; anti-inflammatoire, 0.17; selby, 0.16; escherichia, 0.14; homelessness, 0.14; anti-drogues, 0.14; antidiarrhéique, 0.12; imodium, 0.12; surprescription, 0.10>	$P(e\|f) \leq 0.1$
QT	<drogue, 0.45; medicament, 0.35; consommation, 0.06; relier, 0.03; consommer, 0.02; drug, 0.02; usage, 0.02; toxicomanie, 0.01; substance, 0.01; antidrogue, 0.01; utilisation, 0.01; lier, 0.01; thérapeutique, 0.01; actif, 0.01; pharmaceutique, 0.01>	$P(e\|f)$, 100K
DT	<reflexions, 1; antidrogue, 1; narcotrafiquants, 1; drug, 1; droguer, 0.87; drogue, 0.83; drugs, 0.81; médicament, 0.67; pharmacorésistance, 0.47; pharmacorésistants, 0.44; médicamenteux, 0.36; stupéfiant, 0.34; assurance-médicaments, 0.33; pharmacothérapie, 0.33; amphétamine, 0.18; toxicomane, 0.17; mémorandum, 0.10; toxicomanie, 0.08; architectural, 0.08; pharmacie, 0.07; pharmaceutique, 0.06; thérapeutique, 0.04; substance, 0.01>	$P(f\|e)$, 100K

Table 5.5. Example translations: stems and probabilities with different CLIR methods

The translation of *drug* based on the VLIS database is *stupéfier* with probability 1.0 for both $P(e|f)$ and $P(f|e)$. The lack of alternative translations is a bit surprising, but our default procedure only takes the main translation of a concept of which the main English translation is *drug* and this concept has just a single main translation in French: *stupéfiant*, which is converted into *stupéfier* because all translations are lemmatized. In

this particular instance, the lemmatization procedure is actually unfortunate, since the tagger assumes that *stupéfiant* is a verb form, which it is not. This is an example, that it is quite tricky to extract translation probability estimates from a machine readable dictionary.

5.4.3. The CLEF test collection. To achieve our objective, we carried out a series of experiments on a combination of the CLEF-2000, -2001 and -2002 test collections. [5] This combined test collection consists of documents in several languages (articles from major European newspapers from the year 1994 (CLEF 2000 documents only)), 140 topics describing different information needs (also in several languages) and their corresponding relevance judgements. We only used the English, Italian and French data for the CLIR experiments reported here. The main reason for this limitation was that the IR experiments and Web-based translation models were developed at two different sites equipped with different proprietary tools. We were thus limited to those language pairs for which equivalent normalization steps for both the translation model training and indexing system were available. A single test collection was created by merging the three topic-sets in order to increase the reliability of our results and sensitivity of significance tests. Each CLEF topic consists of three parts: **title**, **description** and **narrative**. An example is given below:

> <**num**> C001
> <**title**> Architecture in Berlin
> <**description**> Find documents on architecture in Berlin.
> <**narrative**> Relevant documents report, in general, on the architectural features of Berlin or, in particular, on the reconstruction of some parts of the city after the fall of the Wall.

We used only the **title** and **description** part of the topics and concatenated these to form the queries. Since the document-sets of the French and Italian part of the CLEF2000 test collection are subsets of the respective document-sets for CLEF2001 and CLEF2002, we based our experiments on the CLEF2000 document set and removed relevance judgements for the additional documents (the SDA set) from the French and Italian `qrel`-files of CLEF2001 and CLEF2002. Table 5.6 lists some statistics on the test collection[6]. The

Document source	Le Monde	LA Times	La Stampa
# documents	44,013	110,250	58,051
# topics	124	122	125
# relevant documents	1189	2256	1878

Table 5.6. Statistics on the test collection

documents are submitted to the same preprocessing (stemming/lemmatization) procedure as we described in section 5.3.1.2. For English and French lemmatization, we used

[5] CLEF=Cross-Language Evaluation Forum, `www.clef-campaign.org`
[6] Topics without relevant documents in a sub-collection were discarded.

the Xelda tools from XRCE[7], which perform morphological normalization slightly differently from the one described in section 5.3.1.2. However, since the two lemmatization strategies are based on the same principle (POS-tagging plus inflection removal), the small differences in morphological dictionaries and POS-tagging had no significant influence on retrieval effectiveness. [8] We also used a Xelda-based morphological normalization procedure for the VLIS-based CLIR experiments involving Italian queries or documents. All runs use a smoothing parameter $\lambda = 0.3$. This value had shown to work well for experiments with several other CLIR collections (Hiemstra et al., 2001b; Kraaij, 2002)

5.4.4. Baseline systems. We decided to have two types of baseline runs. It is standard practice to take a monolingual run as a baseline. Our monolingual baseline run is based on an IR system using document ranking formula 45. Contrary to runs described in Kraaij (2002), we did not use any additional performance enhancing devices, like document length-based priors, pseudo feedback or fuzzy matching in order to focus on just the basic retrieval model extensions, avoiding interactions.

External query translation using Systran served as an additional cross-language baseline, as a reference point for cross-language runs. Notice that the lexical coverage of MT systems varies considerably across language pairs. In particular, the French-English version of Systran is quite good in comparison with other language pairs. We accessed the Web-based version of Systran (December 2002), marketed as "Babelfish" (Yang & Lange, 1998), using the Perl utility `babelfish.pm` and converted the Unicode output to the ISO-latin1 character-set to make it compatible with the Xelda-based morphology.

5.4.5. Results. Table 5.7 lists the results for the different experimental conditions in combination with a translation model pruned with the probability threshold criterion $P > 0.1$ (cf. section 5.3.1.2). For each run, we computed the mean average precision using the standard evaluation tool `trec_eval`. We ran Friedman tests on all the runs based on one particular translation models, because these are the runs we are most interested in; furthermore, one should avoid adding runs that are quite different to a group which is relatively homogeneous, since this would easily lead to a false global significance test. The Friedman test (as measured on the F distribution) proved significant at the $P < 0.05$ level in all cases, so we created equivalence classes using Fisher's LSD method, which are denoted by letters (see table 5.7). Letters are assigned in decreasing order of performance; so if a run is member of equivalence class 'a' it is one of the best runs for that task.

The last four rows of the table provide some additional statistics on the query translation process. For both the forward ($P(t|s)$,fw) and the reverse ($P(s|t)$,rev) translation model, we list the percentage of missed translations (% missed)[9] of unique query terms and the average number of translations (# translations) per unique query term. Table 5.8

[7] `http://www.xrce.xerox.com/competencies/ats/xelda/summary.html`

[8] We have not been able to substantiate this claim with quantitative figures but did analyze the lemmas that were not found in the translation dictionaries during query translation. We did not find any structural mismatches.

[9] Many of the missed translations are proper nouns.

run id	FR-FR	EN-EN	IT-IT	EN-EN.
MONO	0.4233	0.4705	0.4542	0.4705
	EN-FR	FR-EN	EN-IT	IT-EN.
MT	0.3478	0.4043	0.3060	0.3249
QT	a:**0.3760**	a:**0.4126**	a,b:0.3298	a:**0.3526**
DT	a:0.3677	a,b:0.4090	a:**0.3386**	a,b:0.3328
SYN	a:0.3730	b,c:0.3987	a,b:0.3114	b:0.3498
QT-EQ	a:0.3554	a,b:0.3987	c,d:0.3035	b,c:0.3299
QT-MP	a:0.3463	c,d:0.3769	b,c:0.3213	b:0.3221
NAIVE	b:0.3303	d:0.3596	d:0.2881	c:0.3183
% missed fw	9.6	13.54	16.79	9.17
% missed rev	9.08	14.04	15.48	11.31
# translations fw	1.65	1.66	1.86	2.13
# translations rev	22.72	29.6	12.00	22.95

Table 5.7. Mean average precision and translation statistics ($P > 0.1$)

lists the results for the same experimental conditions, but this time the translation models were pruned by taking the n best translation relations according to an entropy criterion, where n=100.000 (100K). Several other similar pruning methods were also tested

run id	FR-FR	EN-EN	IT-IT	EN-EN.
MONO	0.4233	0.4705	0.4542	0.4705
	EN-FR	FR-EN	EN-IT	IT-EN.
MT	0.3478	0.4043	0.3060	0.3249
DT	a:**0.3909**	a:0.4073	a:**0.3728**	a:0.3547
QT	a,b:0.3878	a:**0.4194**	a:0.3519	a:**0.3678**
QT-MP	b:0.3436	b:0.3702	b:0.3236	b:0.3124
SYN	c:0.3270	b:0.3643	b:0.2958	c:0.2808
QT-EQ	c:0.3102	b:0.3725	c:0.2602	c:0.2595
NAIVE	d:0.2257	c:0.2329	d:0.2281	d:0.2021
% missed fw	11.04	14.65	16.06	9.36
% missed rev	10.39	16.81	15.76	10.53
# translations fw	7.04	7.00	6.36	7.23
# translations rev	10.51	12.34	13.32	17.20

Table 5.8. Mean average precision and translation statistics (best 100K parameters)

on the CLEF-2000 subset of the data, e.g. "P>0.01", "P>0.05", "1M parameters", "10K parameters", etc. However, the two cases shown in tables 5.7 and 5.8 represent the best of the two families of pruning techniques. The goal was not to do extensive parameter tuning in order to find the best performing combination of models, but rather to detect some broad characteristics of the pruning methods and their interactions with the retrieval model.

run id	FR-FR	EN-EN	IT-IT	EN-EN.
MONO	0.4233	0.4705	0.4542	0.4705
	EN-FR	FR-EN	EN-IT	IT-EN.
MT	0.3478	0.4043	0.3060	0.3249
QT	a:**0.3468**	a:**0.3055**	a,b:0.3408	a:**0.3141**
DT	b:0.3176	b:0.2801	a:**0.3625**	a:0.3094
SYN	b:0.3097	b:0.2743	b:0.3337	a:0.3082
QT-EQ	b:0.3090	b:0.2920	c:0.3113	a:0.3035
QT-MP	d:0.2503	c:0.2229	d:0.1996	b:0.2634
NAIVE	c:0.2617	c:0.1938	d:0.2062	b:0.2390
% missed fw	2.1	9.7	2.1	4.35
% missed rev	2.1	9.7	2.1	4.35
# translations fw	3.0	4.2	14.1	2.8
# translations rev	3.0	4.2	14.1	2.8

Table 5.9. mean average precision and translation statistics (VLIS)

Table 5.9 presents the results of the experiments with six different CLIR models using translation models estimated on the VLIS lexical database. The relative high proportions of missed translations for FR-EN is due to a small mismatch between the Xelda lemmatizer and the VLIS database. Xelda recognizes complex constructions containing particles like *ainsi que*, which are not listed as lemmas in VLIS. The lexical coverage for content terms seems not significantly lower than for any of the other language pairs. The relatively high number of translations for EN-IT is due to the fact that these translations have been recently added to VLIS and lack the database field *main translation*. A fully integrated VLIS update was not available at the moment of the experiments. Consequently all translations – main and alternative – are included.

5.4.6. Discussion. In this section we will discuss the experimental results in the context of the research questions as formulated in 5.4.1. The questions are each discussed in a separate subsection, except question iv, about the differences between Web-based and VLIS-based translation models. This aspect will be discussed in conjunction with each other research question as far as relevant.

5.4.6.1. *Integrated word-by-word CLIR vs. MT-based CLIR.* Our first observation when examining the data (see also the precision-recall plot in figure 5.1) is that the runs based on the Web-based translation models perform comparable to or better than the MT run. Sign tests showed that there was no significant difference between the MT and QT runs for EN-FR and FR-EN language pairs. The QT runs were significantly better at the P=0.01 level for the IT-EN and EN-IT language pairs. This is a very significant result, particularly since the performance of CLIR with Systran has often been among the best in the previous CLIR experiments in TREC and CLEF. These results show that the Web-based translation models are effective means for CLIR tasks.

The best CLIR performance with Web-TM varies from 74.1% to 93.7% of the monolingual run. This is within the typical range of CLIR performance. More generally, this

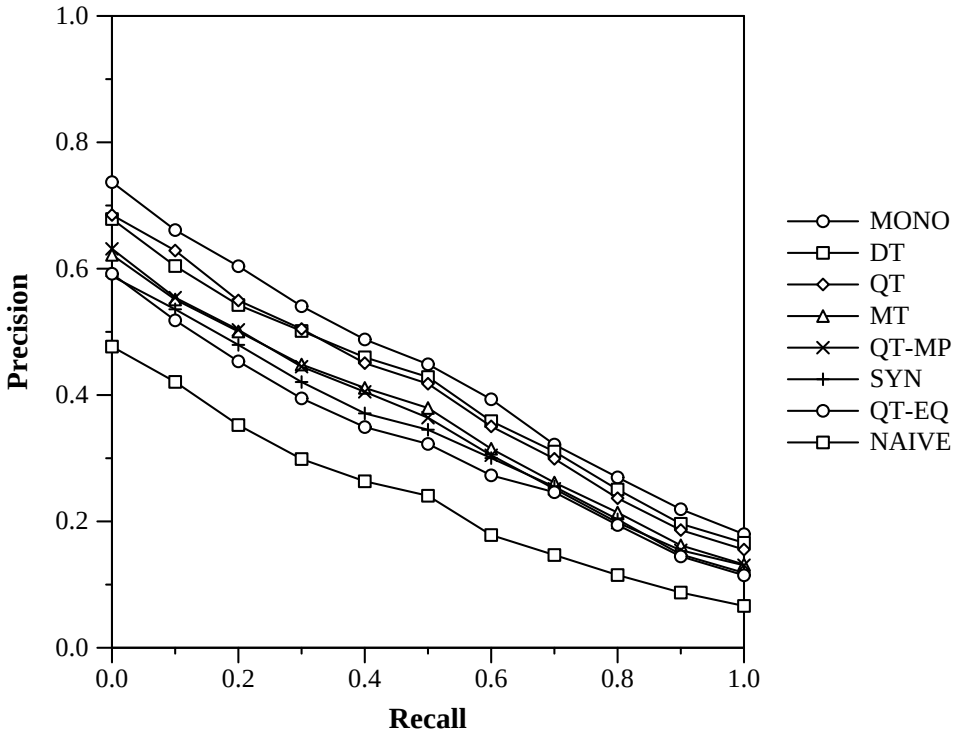

Figure 5.1. precision-recall plot of the best performing EN_FR runs with 100K translation models

research successfully demonstrates the enormous potential of parallel Web pages and Web-based MT.

We cannot really compare performance across target languages, since the relevant documents are not distributed in a balanced way; some topics even do not contain a single relevant document in a particular sub-collection. We can, however, compare methods within a given language pair.

For the VLIS-based translation models, the pattern is similar, although their effectiveness seems to be a little lower than the Web based runs. Effectiveness is also lower than Systran for most language pairs. We hypothesize that the disappointing performance of the VLIS based runs is due to the poor probability estimates, and not to a lack of dictionary coverage, since the coverage of VLIS is higher than the Web-based translation models (cf tables 5.2 and 5.3). A further analysis of this issue is reported in sections 5.5.7.1 and 5.5.5.

5.4.6.2. *Comparison of query model translation (QT), document model translation (DT) and translations modelled as synonyms (SYN).* Our second question in section 5.4.1 concerned the relative effectiveness of the QT and DT models. We will first discuss the Web-based models

Web-based models.
The experimental results show that there is no clear winner; differences are small and not significant. There seems to be some correlation with translation direction, however: the QT models perform better than DT on the X-EN pairs and the DT models perform better on the EN-X pairs. This might indicate that the $P(e|f)$ and $P(e|i)$ translation models are more reliable than their reverse counterparts. A possible explanation for this effect could be that the average English sentence is shorter than a French and Italian sentence. The average number of tokens per sentence is 6.6/6.9 and 5.9/6.9 for EN/FR and EN/IT corpora respectively. This may lead to more reliable estimates for $P(e|f)$ and $P(e|i)$ than the reverse. However, further investigation is needed to confirm this, since differences in morphology could also contribute to the observed effect. Still, the fact that QT models perform just a good as DT models in combination with translation models is a new result.

We also compared the QT and DT methods to the synonym-based approach of (Pirkola, 1998). Both the QT and DT model were significantly more effective than the synonym-based model (SYN). The latter seems to work well when the number of translations is relatively small, but cannot effectively handle the large number of (pseudo)-translations as produced by our 100K translation models. The synonym-based model usually performs better than the models based on query translation with uniform probabilities, but differences are not significant in most cases.

VLIS-based models.
For the VLIS-based models, there is a significant difference between QT and DT for the first two language pairs. For English-Italian, the DT is better than QT. Possible explanations could be that (i) QT is not well equipped to handle many translations per word when the relative probability estimates are poor ii) DT is better equipped to handle many (poorly weighted) translations. We will try to do some further investigations regarding this issue in section 5.5. Differences between DT and SYN (the Pirkola model) are not significant except for English-Italian. We hypothesize that the fact that SYN model has an acceptable performance in comparison with the probabilistic models is due to the low average number of translations. When the average number of translations is increased, relative performance of SYN is lower than for the fully weighted models.

5.4.6.3. *Query expansion effect.* In the introduction we argued that using just one translation (as MT does) is probably a suboptimal strategy for CLIR, since there is usually more than one good translation for a term. Looking at probabilistic dictionaries, we have also seen that the distinction between a translation and a closely related term cannot really be made on the basis of some thresholding criterion. Since it is well known in IR that adding closely related terms can potentially improve retrieval effectiveness, we hypothesize that adding more than one translation would also help. The experimental results confirm this effect. In all but one case (EN-FR, $P > 0.1$) using all translations (QT) yielded significantly better performance than choosing just the most probable translation (QT-MP). For the $P > 0.1$ models, the average number of translations in the forward direction is only 1.65, so the potential for a query expansion effect is limited, which could explain the non-significant difference for the EN-FR case. The differences between QT and QT-MP are considerable larger for the VLIS-based runs. Since the runs based on

the most probable translation based on VLIS are some 25-30% below the most probable translation based on the Web corpus and the coverage of the VLIS dictionaries is quite good, we can conclude that translation probability estimates based on VLIS are inferior to the corpus-based estimates. This really hurts performance of the QT-MP VLIS runs.

Unfortunately, we cannot say whether the significant improvement in effectiveness of runs based on more translations is mainly due to the fact that the probability of giving at least one good translation (which is probably the most important factor for retrieval effectiveness (Kraaij, 2002; McNamee & Mayfield, 2002)) is higher for QT or indeed to the query expansion effect. A simulation experiment is needed to quantify the relative contributions. Still, it is of great practical importance that more (weighted) translations can enhance retrieval effectiveness significantly. In section 5.5.6 we will present some additional experimental results, which prove that there is an effective query expansion effect. In section 5.5.5 we will investigate why the Web-based QT-MP run is about as good as the VLIS based QTrun.

5.4.6.4. *Pruning & weighting.* A related issue is the question of whether it is more important to prune translations or to weight them. Grefenstette (cf. section 5.1.4) originally pointed out the importance of pruning and weighting translations for dictionary-based CLIR. Pruning was seen as a means of removing unwanted senses in a dictionary-based CLIR application. Our experiments confirm the importance of pruning and weighting, but in a slightly different manner. In a CLIR approach based on a Web translation model, the essential function of pruning is to remove spurious translations. Polluted translation models can result in a very poor retrieval effectiveness. As far as sense disambiguation is concerned, we believe that our CLIR models can handle sense ambiguity quite well. Our best performing runs, based on the 100K models, have on average seven translations per term! Too much pruning (e.g. best match) is sub-optimal. However, the more translation alternatives we add, the more important their relative weighting becomes.

We have compared weighted translations (QT) with uniform translation probabilities (QT-EQ). In each of the twelve comparisons (four language pairs, three sets of translation models), weighting results in a improved retrieval effectiveness. The difference is significant in nine cases. Differences are not significant for the $P < 0.1$ EN-FR and FR-EN translation models. We think that for the Web-based models, this is due to the small average number of translations; a uniform translation probability will not differ radically from the estimated translation probabilities. For the VLIS models, there is a relative difference of around 10 % for the EN-FR and EN-IT language pairs. The relative difference is much smaller for the reverse pairs and not significant for IT-EN in particular. We do not have a good explanation for these differences across pairs. Probability estimates for VLIS-based models are poor, so the effectiveness of CLIR runs based on those models might be largely determined by particularities of the individual lexical databases.

The importance of weighting is most evident when the 100K Web-based translation models are used. These models yield seven translations on average for each term. The CLIR models based on weighted translations are able to exploit the additional information and show improved effectiveness w.r.t. the $P < 0.1$ models. The performance of unweighted CLIR models (QT-EQ and SYN) is seriously impaired by the higher number of translations.

The comparison of the naive dictionary-like replacement method, which does not involve any normalization for the number of translations per term (NAIVE), with QT-EQ shows that normalization (i.e. a minimal probabilistic embedding) is essential, especially when the average number of translation per term is high. The NAIVE runs have the lowest effectiveness of all variant systems (with significant differences). For the Web-based translation models it seems better to select just the one most probable translation rather than taking all translations unweighted. For the VLIS-based translation models, the NAIVE method is roughly equally as effective as the QT-MP method, this probably means that the additional gain of adding more translations (increasing the probability of having at least one good translation) is cancelled out by the poor embedding of translation into the retrieval model. Most probability mass is assigned to terms with many translations, which are usually less discriminating terms.

5.4.6.5. *Robustness.* We pointed out in the previous section that the weighted models are more robust, in the sense that they can handle a large number of translations. We found however that the query model translation method (QT) and the document model translation method (DT) display a considerable difference in robustness to noisy translations (which are present in the Web-based models). Initially we expected that the DT method (where the matching takes place in the source language) would yield the best results, since this model has previously proven to be successful for several quite different language pairs, e.g. European languages, Chinese and Arabic using parallel corpora or dictionaries as translation devices (McNamee & Mayfield, 2001; Xu et al., 2001; Hiemstra et al., 2001a).

However, our initial Web-based DT runs yielded extremely poor results. We discovered that this was largely due to noisy translations from the translation models (pruned by the $P < 0.1$ or 100K method). There are many terms in the target language, which occur very rarely in the parallel Web corpus. The translation probabilities for these terms (based on the most probable alignments) are therefore unreliable. Often these rare terms (and non-words like `xc64`) are aligned with more common terms in the other language and are not pruned by the default pruning criteria ($P > 0.1$ or best 100K parameters), since they have high translation probabilities. This especially poses a problem for the DT model, since it includes a summation over all terms in the target language that occur in the document and have a non-zero translation probability. We devised a supplementary pruning criterion to remove these noisy translations, discarding all translations for which the source term has a marginal probability in the translation model which is below a particular value (typically $10^{-6} - 10^{-5}$). Later we discovered that a simple pruning method was even more effective: discard all translations where either the source or target term contains a digit. The results in Tables 5.7 and 5.8 are based on the latter additional pruning criterion. The QT approach is less sensitive to noisy translations arising from rare terms in the target language, because it is easy to remove these translations using a probability threshold. We deduce that extra care therefore has to be taken to prune translation models for the document model translation approach to CLIR.

We also experimented with using forward probabilities $P(t|s)$ as translation "weights" in a DT model. This corresponds to assuming that $P(t_j|s_i) = P(s_i|t_j)$, which obviously

does not hold. Still this approach yielded quite good results, whereas we initially encountered some problems with the $P(s|t)$ models, which introduced a lot of noise. We think that using the forward probabilities $P(t|s)$ provides, unintentionally, an effectively pruned translation set for the highly noise sensitive DT model (67). This provides additional evidence for the fact that for the DT model, pruning spurious translations is more important than weighting translations.

5.5. Experiments II

In this section we report on several additional experiments and analyses carried out to verify some hypotheses that came up after analysing the first set of experiments.

First we investigate the interaction between retrieval effectiveness and the number of translation alternatives for the different CLIR models that we presented earlier in some more detail. This study demonstrates the importance of proper weighting of translation alternatives. We also investigate the relative contribution of using part-of-speech disambiguation for the lexical lookup process.

Subsequently we will show that retrieval performance can be improved further by combining models or resources. It is also very well possible to combine language pairs for an inter-lingual CLIR approach.

We designed additional experiments in order to get a better idea of the relative importance of weighting versus extended lexical coverage (section 5.5.5) and to demonstrate that translation models can be used for effective query expansion.

Finally, we will present a limited query-by-query analysis to gain understanding which factors play a role in performance differences of runs based on different translation resources (e.g. word-by-word translation vs. Systran) and bilingual CLIR runs versus monolingual runs.

5.5.1. Varying the pruning threshold. Since we have seen that there is a strong interaction between the average number of translations and retrieval performance for some of the CLIR methods, we did some additional experiments with a more controlled variation of the level of pruning. We applied the best N parameters pruning method (based on the IBM1-gains criterion), with N=10K, 100K and 1M.

Results of experiments with several CLIR models are presented in Table 5.10 and figure 5.2. There is a clear division between two groups of CLIR models: on the one hand there are the QT and DT models of which the performance increases with the number of parameters (=translation relations) in the translation model, performance is slightly lower for 1M parameters, which shows the necessity of some pruning. On the other hand there are the QT-EQ, SYN and NAIVE models of which the performance is seriously hurt when more translations are added. These models do not benefit from more translations (100K) and seriously break down for larger query expansions. The average number of translations per term is almost linearly related to the number of parameters in the translation model, which is what we expected. The plot also confirms the main conclusion of Franz et al. (2001), who state that query out of vocabulary rate (OOV, which is equivalent to our % missed translations) is a simple (inverse) estimator of the utility of translation models for CLIR systems (in terms of mean average precision).

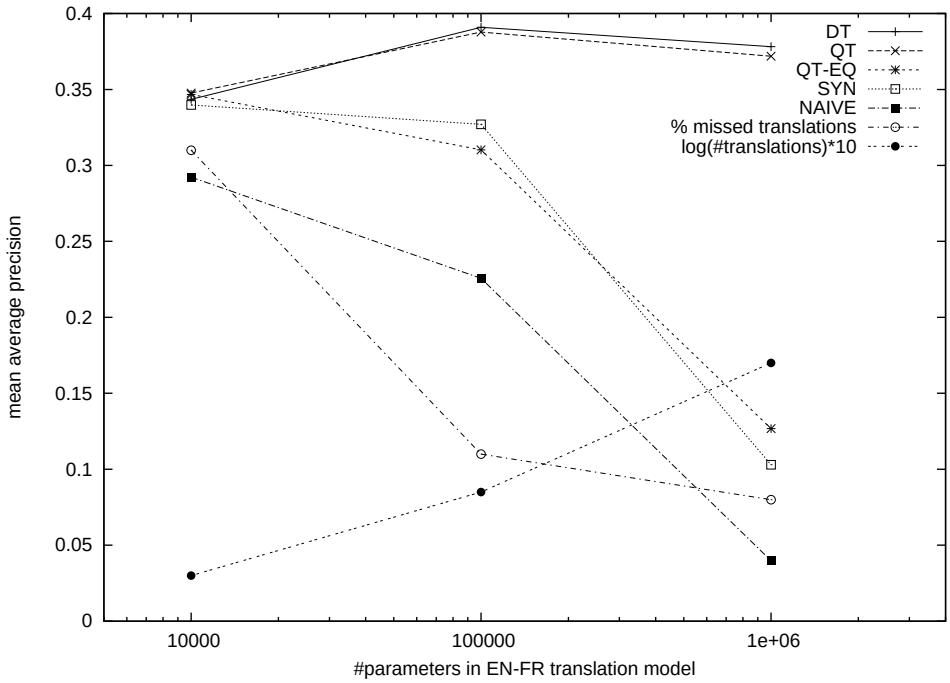

Figure 5.2. Interaction between average number of translations, the number of terms without a translation and the mean average precision of several CLIR models

Performance does increase with decreasing OOV rate, although we found that some pruning is necessary, since the gain of a reduced OOV rate is levelled off by the amount of added noisy translations.

	10K		100K		1M	
run id	EN-FR	FR-EN	EN-FR	FR-EN	EN-FR	FR-EN
DT	0.3435	0.3362	0.3909	0.4073	0.3782	0.4109
QT	0.3476	0.3303	0.3878	0.4194	0.3719	0.4088
QT-EQ	0.3467	0.3321	0.3102	0.3725	0.1268	0.1147
SYN	0.3398	0.3249	0.3270	0.3643	0.1030	0.1125
NAIVE	0.2923	0.2860	0.2257	0.2329	0.0398	0.0315
% missed fw	31.25	31.75	11.04	14.65	8.20	10.65
% missed rev	29.78	35.39	10.39	16.81	8.20	10.65
# translations fw	1.98	1.82	7.04	7.00	58.68	62.24
# translations rev	2.42	2.24	10.51	12.34	200.87	237.22

Table 5.10. Mean Average Precision and translation statistics (best 10K, 100K and 1M parameters)

5.5.2. Different constraints for VLIS lookup. During the development of the lexical lookup procedure of the VLIS lexical database we had added several constraints in order to improve results. The constraints were based on experimentation (on different datasets), heuristics and the aim to use available linguistic knowledge in a sound and effective way.

We carried out additional experiments with the VLIS-based models, to investigate the effect of individual constraints in the lexical lookup procedure on retrieval effectiveness (Table 5.11). Two alternative lookup methods are reported:

All translations (QT-ALL and DT-ALL): Instead of selecting just main translations, all translations (main and alternative) were retrieved. We conjectured that this operation might increase effectiveness, since additional translations improved effectiveness for the Web-based models. There was hardly an effect for DT but the QT method was hurt by the addition of alternative translations. We hypothesize that the alternative translations are less common, but that this is not reflected in the estimation procedure which in absence of quantitative data does assign equal probabilities.

No part-of-speech constraint: We also investigated the influence of using POS information as a constraint in the lexical lookup. Table 5.11 shows that using or not using such a constraint (QT vs. QT-NO and DT vs. DT-NO) hardly makes a difference. There are several explanations for this small difference, first of all, POS ambiguity does not occur very frequently, secondly the POS constraint in lexical lookup sometimes improves and sometimes hurts average precision. These effects cancel out on average.

run id	EN-FR	FR-EN	EN-IT	IT-EN.
MONO	0.4233	0.4705	0.4542	0.4705
MT	0.3478	0.4043	0.3060	0.3249
QT	**0.3468**	**0.3055**	0.3408	**0.3141**
QT-ALL	0.3135	0.2815		0.2893
QT-NO	0.3435	0.3041	0.3290	0.3120
DT	0.3176	0.2801	**0.3625**	0.3094
DT-ALL	0.3181	0.2825		0.2935
DT-NO	0.3177	0.2832	0.3614	0.3094

Table 5.11. Alternative lookup procedures for VLIS

5.5.3. Combination runs. Since the pruned forward and reverse Web translation models yield different translation relations (cf. table 5.5), we hypothesized that it might be effective to combine both. Instead of combining the translation probabilities directly we chose to combine the results of the QT and DT by interpolation of the document scores. Results for combinations based on the 100K models are listed in table 5.12. Indeed, for all the language pairs, the combination run improves upon each of its component runs. The most plausible explanation is that each component run can compensate for missing translations in the companion translation model. We did some supplementary runs based on simple linear interpolation (with interpolation parameter 0.5) of document

nr	run id	EN-FR		FR-EN		EN-IT		IT-EN	
1	MONO	0.4233		0.4705		0.4542		0.4705	
2	MT	0.3478	(82%)	0.4043	(86%)	0.3060	(67%)	0.3249	(69%)
3	Web DT	0.3909	(92%)	0.4073	(86%)	0.3728	(82%)	0.3547	(75%)
4	Web QT	0.3878	(92%)	0.4194	(89%)	0.3519	(75%)	0.3678	(78%)
5	3+4	0.4042	(96%)	0.4273	(91%)	0.3837	(84%)	0.3785	(80%)
6	VLIS QT	0.3468	(82%)	0.3055	(65%)	0.3408	(75%)	0.3141	(67%)
7	VLIS DT	0.3176	(75%)	0.2801	(60%)	0.3625	(80%)	0.3094	(66%)
8	6 + 7	0.3410	(81%)	0.3016	(64%)	0.3642	(80%)	0.3210	(68%)
9	2+5	0.4106	(97%)	**0.4366**	(93%)	0.3924	(86%)	0.3932	(84%)
10	2+ 8	0.3854	(91%)	0.4082	(87%)	0.3928	(86%)	0.3694	(79%)
11	10 + 5	**0.4208**	(99%)	0.4278	(91%)	**0.4254**	(94%)	**0.4139**	(88%)

Table 5.12. Mean Average Precision of combination run, compared to baselines

scores for runs with different translation resources. We expected that especially the combinations where different translation resources are combined, will yield improved performance. This is indeed the case, for three out of four language pairs, results improve upon each combination step, e.g. ((VLIS+MT)+Web)>(VLIS+MT)>VLIS. For FR-EN, the (Web+MT) run is better, since the results for the VLIS run are too inferior in comparison with the Web and Systran run. A much more extensive study of combination algorithms is possible, but falls beyond the scope of this thesis. What we want to conclude here is that it is possible to reach almost the same level of retrieval effectiveness as a monolingual run, using a probabilistic word-by-word translation model and a combination of translation resources.

5.5.4. Transitive Translation. An important advantage of CLIR based on parallel web corpora is that it will lead to resources for many more language pairs than covered by commercial MT systems. For most supported pairs English will be one the two languages, since it is the dominant language in international business and science. Therefore we hypothesized already that English could be used as a pivot language to maximize the number of different language pairs for which CLIR resources are available. In the following section we will report some preliminary experiments that were carried out with transitive approaches to CLIR based on parallel Web corpora and MRD's: FR-EN-IT and IT-EN-FR. The different approaches are illustrated in Figure 5.3. We evaluated three different ways (76- 78) to implement such a transitive approach, the first two alternatives use the convolution operation to combine two language models:

$$P(t_i|Q_s) \approx \sum_{k}^{I} \sum_{i}^{S} P(t_j|v_k) P(v_k|s_i) P(s_i|Q_s) \tag{76}$$

This is a variant of model (65) - the QT model - based on a transitive estimate of $P(t|Q_s)$ - where v_k is a term in the inter-lingual language and I is the vocabulary size of the

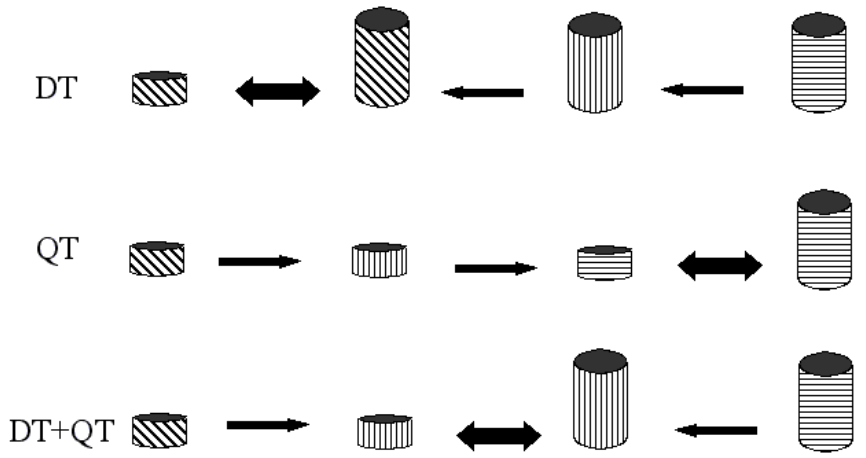

Figure 5.3. Schematic view of the three different ways to use a pivot language for CLIR. The small cylinders represent query models. The large cylinders represent document models. The source, pivot and target language are each represented by a different line pattern. The double arrow represents the matching operation between query and document model. The single arrow represents a translation step.

interlingua.

$$P(s_i|D_t) \approx \sum_{k}^{I} \sum_{j}^{T} P(s_i|v_k) P(v_k|t_j) P(t_j|D_t) \tag{77}$$

This is a variant of model (67) - the DT model - based on a transitive estimate of $P(t|D_t)$.

(78)
DT+QT:

$$CER(Q_s; C_t, D_t) = \sum_{k=1}^{I} \sum_{i=1}^{S} P(v_k|s_i) P(s_i|Q_s) \log \frac{\sum_{j=1}^{T} P(v_k|t_j)((1-\lambda)P(t_j|D_t) + \lambda P(t_j|C_t))}{\sum_{j=1}^{T} P(v_k|t_j) P(t_j|C_t)}$$

The last model is a variant where both the query and the documents are translated and matching takes thus place in the inter-lingual language. Table 5.13 presents the results of the experiment. We have provided two baselines: a monolingual run and a run using the synonym operator, which has been used by most other authors working on transitive translation (Ballesteros, 2000; Lehtokangas & Airio, 2002). With a performance ranging between 67% and 80% with respect to the monolingual baseline, the results of all three methods are at least at a comparable level as those reported in (Franz et al., 2000) and do significantly outperform the SYN baseline. It is perhaps not surprising anymore that the LM-based methods perform better than the SYN baseline, since the SYN based model cannot leverage the probabilities of the translation alternatives. All translation alternatives

	IT-EN-FR		FR-EN-IT	
monolingual baseline	0.4233		0.4542	
bilingual baselines:				
QT (EN→{FR|IT})	0.3878	(-8%)	0.3519	(-23%)
DT (EN←{FR|IT})	0.3909	(-8%)	0.3728	(-18%)
transitive runs:				
SYN(Pirkola)	b:0.1469	(-65%)	b:0.2549	(-44%)
QT (target match)	a:0.2924	(-31%)	a:0.3287	(-28%)
DT (source match)	a:**0.3149**	(-26%)	a:**0.3598**	(-21%)
QT+DT (pivot match)	a:0.2866	(-32%)	a:0.3361	(-26%)
% missed qt	14.5		17	
% missed dt	16		20	
% missed qt+dt	11		11	
# translations qt	9.6		9.4	
# translations dt	84.0		123	
# translations qt+dt	55.0		97	

Table 5.13. Results of transitive CLIR runs based on a combination of 100K models (mean average precision)

are equally probable in this approach and many translation alternatives amounts thus to high ambiguity. Recent work by Ballesteros confirms this weakness of the SYN based approach Ballesteros & Sanderson (2003). This weakness can be compensated by a probabilistically motivated version of weighted structured queries Darwish & Oard (2003), but the resulting model is less transparent than our cross-entropy based approach where translation is a part of the model. The QT , DT, and QT+DT methods have a slightly different performance, but differences are not consistent across both language pairs. We performed a Friedman significance test. The overall test showed significant differences for both language pairs. Pairwise comparions[10] showed that there were no significant differences between the QT, DT and QT+DT methods for both IT-EN-FR and FR-EN-IT. All the LM-based methods are significantly better than the SYN baseline at the 0.01 level. We think that the differences are due to lexical mismatches between the constituting models. For a CLIR run on the Italian test collection using French queries and the QT model with English as an interlingua, the first model maps the French query model into an English query model, whereas the second query model maps the English query model in an Italian query model. Not all terms that can be translated by one model, have a non zero translation probability in the other model. An important reason for this imbalance is the fact that the models are trained on different parallel corpora of different sizes. Since the translation models themselves are not symmetric, this will result in differences between methods. A comparison of the number of missed translations with the runs based on just a single translation step (table 5.8) shows that this is a serious effect. The EN-IT statistical translation dictionary is substantially smaller than the EN-FR translation

[10] Equivalence classes are denoted by letter prexises in table 5.13.

dictionary (about 35K vs. 50K entries). This explains why mean average precision is hurt more by going from EN-FR to IT-EN-FR than going from EN-IT to FR-EN-IT.

The data seem to suggest a positive correlation between the number of translations and the mean average precision (with the exception of QT+DT for IT-EN-FR). Indeed, this seems plausible, since more translation relations would help to provide a more robust mapping of a language model from one language to another. However, our bilingual experiments presented in section 5.4 using the same test collection do not show this correlation. In the bilingual experiments, translation effectiveness seemed dependent on the relative verbosity of the languages involved. Translation from the more verbose language to the less verbose language (e.g. French → English) was more effective. Moreover, the experiment with pivoted translation using symmetric data from MRD's as reported below does not suggest a correlation.

We repeated the experiment with translation models based on VLIS although we used the "direct" estimation method of (72) and (73) instead of the convolution approach. Results are presented in table 5.14. The performance of the VLIS-based runs on Italian

	IT-NL-FR		FR-NL-IT	
monolingual baselines	0.4233		0.4542	
transitive runs:				
SYN	b:0.3421	(-19%)	c:0.3171	(-30%)
QT	a:**0.3542**	(-16%)	a:0.3171	(-30%)
DT	a:0.3468	(-18%)	a,b:**0.3391**	(-25%)
QT+DT	a:0.3473	(-18%)	b,c:0.3080	(-32%)
% missed qt	6.2		10	
% missed dt	6.2		10	
% missed qt+dt	6.2		10	
# translations qt	3.4		6.4	
# translations dt	3.4		6.4	
# translations qt+dt	4.6		10.6	

Table 5.14. Transitive translation based on different VLIS-based models (performance difference with EN-FR and EN-IT respectively in brackets)

and French documents does not differ dramatically from the results based on English queries presented in table 5.9. This can hardly come as a surprise, since all the runs use the inter-lingual Dutch representation as a pivot language and thus do not differ in a principal way. Again there is no clear sign that either of the models QT, DT or QT+DT is clearly superior over the other. This time, we can directly compare QT and DT since the number of translations is exactly the same. Sign-test show that there is no significant difference between QT, DT and QT+DT for the IT-NL-FR runs, but QT and DT are significantly better than SYN. For the FR-NL-IT runs, the DT run is significantly better (sign test at 0.05) than the other methods. These results were confirmed by the Friedman signficance test. Since the order of CLIR models based on retrieval effectiveness is completely different for the IT-NL-FR runs, we will not draw any strong conclusions. There is a strong interaction between the translation resource, the query set, the document

collection and retrieval performance. Further research is needed to explore the nature of this interaction, e.g. by performing a query-by-query analysis.

After two sets of experiments with CLIR using a pivot language and translation models in various embeddings we conclude that the particular embedding is not so important (as long as it is probabilistically sound), but that it is important to combine translation resources with a comparable lexical coverage. Both Web-based and dictionary-based transitive CLIR methods yielded performances between 63-83% of a monolingual setting.

5.5.5. Web-based QT-BM: better translations or better weighting? Since the results of taking just the best translation from the Web based models, produced results comparable to taking all (pseudo) weighted translations from the lexical database (for EN-FR and IT-EN) we were curious whether the Web was especially good in finding extra translations or that the VLIS-based runs were not optimal because of the poor weighting strategy. There is a strong indication for the latter reason, since the best match VLIS-based runs perform very disappointing. Nevertheless we devised a test to find the correlation between the proportion of Web-based translations not found in VLIS and the performance difference between the Web QT-MP run and the VLIS QT-ALL run. We selected the QT-ALL run (cf. section 5.5.2) since it contains all translations for a particular term that exist in VLIS. For each topic, we calculated the proportion of terms from the QT-MP translation that was not found in the VLIS translations and plotted this fraction against the difference in average precision between both runs. The result is presented in a scatter-plot: figure 5.4. At the left side of the plot, the fraction is zero (i.e. all translations based on the Web dictionary are also listed in the VLIS-based translation). There are many topics where the fraction is zero and there is a lot of variation in the performance difference. This is an indication that the size of the fraction does not account for the variance in performance difference across topics. The scatter-plot shows that there is a small correlation between the fraction of Web translations not found in VLIS and a positive performance difference. The Pearson product moment correlation coefficient is $\rho = 0.08$, confirming that the fraction and performance difference are almost independent. Therefore we conclude that the fact that the Web translation is based on a better weighting more than compensates for the fact that it uses just a single translation (i.e. the probability of providing at least one good translation is lower than the VLIS QT-ALL run).

5.5.6. Improving monolingual translation by cross-lingual expansion. In section 5.4.6.3, we concluded that it was difficult to prove that the good results of using more than one translation are mainly due to a query expansion effect or due to the increased probability of at least one good translation (reducing the query OOV rate). Therefore we designed an experiment where we can precisely measure the query expansion effect, namely a setting where we translate a query (or document) to another language and back. This will result in an expanded query (or document) consisting of the original terms plus new terms that are added by the statistical translation dictionaries. Now, if the expanded queries will have improved effectiveness with respect to the original queries, we can conclude that the translation models are an effective resource for query expansion.

We performed a series of experiments using the models for transitive translation discussed in the previous section. The QT and DT methods effectively perform query and

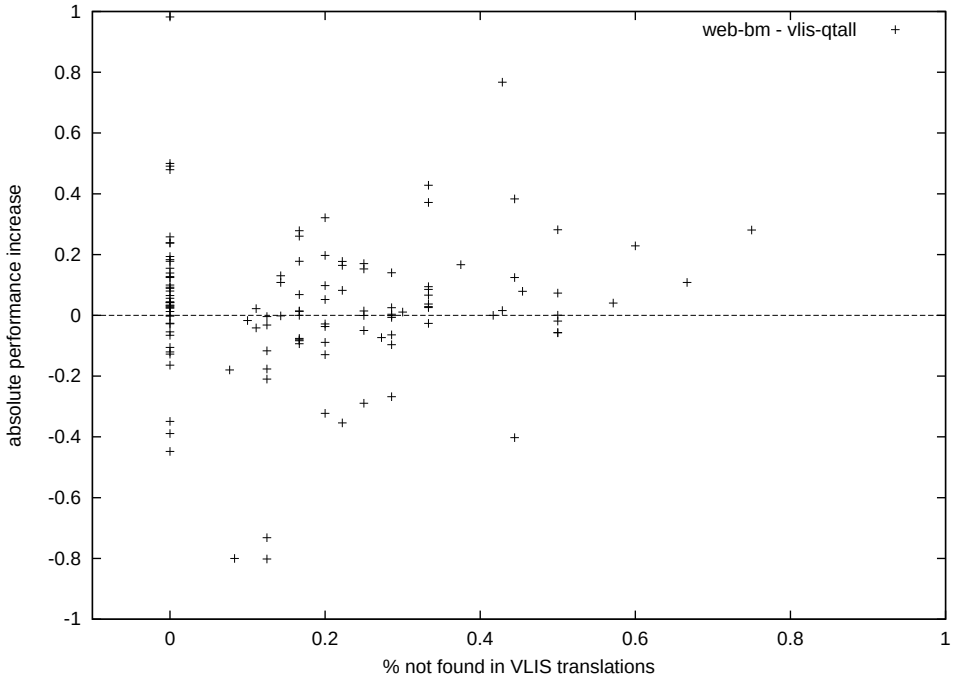

Figure 5.4. Performance increase of the Web-based QT-MP run versus the VLIS-based QT-ALL run as a function of the proportion of Web-based translations NOT found between the VLIS translations.

document expansion via a statistical thesaurus trained on a parallel corpus or translation dictionary (see also (Xu et al., 2002b)) in the following way:

$$P(s_i|s_i) = \sum_{j}^{T} P(s_i|t_j)P(t_j|s_i) \tag{79}$$

where s_i is a term in the source language and t_j is a term in the target language, which is used as a pivot. The QT-DT method will "translate" both query and document language models into the target language, where matching will take place.

We can even improve upon these runs, by combining them with the baseline run, which means that we stress the original terms. We combined the expanded run with the baseline run, using a simple interpolation approach, where the RSV of a document in the combined run is the weighted average of the component runs. We chose $\alpha = 0.5$ for the interpolation parameter. Table 5.15 presents the results of experiments with English, French and Italian, using translation models trained on Web corpora and estimated on VLIS. Overall, we can conclude that the combination of translation models and probabilistic CLIR models does result in effective query expansion both for Web and VLIS-based translation resources i.e. an improved mean average precision, albeit that runs have to be combined with baseline runs (which is a standard operation in relevance

run id	'pure' run	combination ($\alpha = 0.5$)	% rel. diff	rel ret.
EN MONO	0.4705			2120 (2256)
QT(Web via FR)	0.4656	++0.4841	2.9	2151
DT(Web via FR)	0.4555	++0.4837	2.8	2154
QT+DT (Web via FR)	0.4747	++0.4850	3.1	2162
QT(VLIS via "NL")	0.4569	0.4732	0.6	2126
DT(VLIS via "NL")	0.4555	0.4705	0	2133
QT+DT (VLIS via "NL")	0.4404	0.4638	-1.4	2121
FR MONO	0.4233			1821 (1878)
QT(Web via EN)	0.4331	++0.4498	6.3	1820
DT(Web via EN)	0.4215	++0.4497	6.3	1835
QT+DT (Web via FR)	0.4363	++0.4507	6.5	1832
QT(VLIS via "NL")	0.4187	0.4330	2.3	1825
DT(VLIS via "NL")	0.4247	0.4289	1.3	1811
QT+DT (VLIS via "NL")	0.4288	0.4356	2.9	1810
IT MONO	0.4542			1163 (1189)
QT(Web via EN)	0.4311	+0.4777	5.2	1155
DT(Web via EN)	0.4389	0.4654	2.5	1150
QT+DT (Web via FR)	0.4690	++0.4740	4.4	1155
QT(VLIS via "NL")	0.4256	+0.4683	3.1	1165
DT(VLIS via "NL")	0.4468	++0.4628	1.9	1161
QT+DT (VLIS via "NL")	0.4509	++0.4702	3.5	1160

Table 5.15. Mean average precision of monolingual query expansion using a pivot language, mixed with a baseline monolingual run and the relative improvement w.r.t. a baseline run. '++' marks significance at the 0.01 level, '+' marks significance at the 0.05 level. The last column presents the total number of relevant documents retrieved

feedback). The majority of the combination runs are significantly[11] better than their respective baselines. The performance increase is sometimes mostly based on an overall higher recall (English runs) and sometimes more on improved precision (Italian and French runs).

On a more detailed scale, there seem to be no systematic differences between the QT, DT and QT+DT models. Observed differences are most probably related to differences in translation models and their interaction with the queries and documents. On the other hand it seems that the largest performance increase is reached by the runs using the Web-based translations. This does not come as a surprise anymore (cf. section 5.4.5): firstly, the Web-based translations are based on word associations mined from corpora and thus are a richer resource, and secondly the associated weights (translation probabilities) have more reliable estimates.

Of course, the results in this section are not meant to be interpreted as an upper-bound for performance increase using a (parallel) corpus for expansion. It could very

[11] Significance was tested by Sign tests.

well be that simple monolingual query expansion on the corpus that was used to train the translation models, would result in even better results. What is shown is that our probabilistic CLIR models do have a real potential for query expansion when multiple weighted translations are kept.

5.5.7. Query-by-query analysis. A well-known fact in IR is the high variance in retrieval performance (usually measured as average precision) across topics. A large set of topics is therefore required in order to support statistical inferencing, i.e. making statements about differences in performance between different methods. The large variation is due to differences in topic "hardness" and differences in preciseness of the query formulation. Averaging over a large set of topics, smoothes out these differences and focuses on the global picture. However, averaging across topics also hides a lot of detail. It is not so evident which effect(s) cause an increase or decrease in mean average precision. This is especially the case when comparing runs where there is less control on the experimental variables. A method to get more insight into what effects are taking place on a qualitative level is a so-called query-by-query analysis, which inspects what happens in detail for a certain query. In our case we analysed the differences between different (translated) queries. We carried out a query by query analysis for the comparison of different translation resources and to gain insight in differences between bilingual and monolingual runs.

5.5.7.1. *Qualitative differences between translation resources.* We have carried out a query-by-query analysis for a limited set of topics (the CLEF 2001 topic collection). In our discussion in section 5.4.6, we concentrated on a comparison of different methods to embed a translation resource into the retrieval model. The query-by-query analysis was limited to those topics where differences in retrieval performance between runs was high. We performed such a comparison for Systran versus Web-based translation models and Systran versus VLIS-based translation models.

FR-EN: Systran versus Web parallel corpus. A striking result is that the Web-based runs perform significantly better than the Systran-based Babelfish service. We looked at some topics with marked differences in average precision in order to get a better idea which factors play a role. Firstly, the topics where the web corpus run performs better: in topic 47 (+0.55), Systran lacks the translation of Tchétchénie (Chechnya); in topic 58 (+0.46), Systran translates *mort* and *mourir* with *died* and *die*, whereas the web corpus has the additional concepts of *death* and *dead*; topic 82 (about IRA attacks, +0.4) Systran translates *l'IRA* erroneously by *WILL GO*, the corpus-based translation brings in the related term *bomb* as a translation of attack. Secondly, the topics where Systran performs much better: topic 65 (-0.39) the corpus translations of *trésor* are *treasury* and *board*, which would be a fine phrase translation. In this context however, *trésor* does not have the financial meaning and because our system does not recognise phrases, *treasury* and *board* are used as separate query terms, which has the effect that the much more frequent term *board*, brings in a lot of irrelevant documents (the topic is about treasure hunting) . Topic 75 (-0.98) suffers from a wrong interpretation of the word *sept*, which is translated by *sept (September)* and by *7*, the latter term is discarded by the indexer. The month abbreviation retrieves a lot of irrelevant documents, resulting in a low position of the single relevant document; in topic 80 (about hunger strikes) *faim* is translated

both by *hunger* and by *death*. *Death* might be a related term in some cases, but it also retrieves documents about strikes and death, hurting precision; topic 89 talks about an *agent immobilier*, Systran produces the correct translation *real estate agent*, but the corpus-based translation is *officer* and *document* as additional translations for *agent*. Here, the phrase translation of Systran is clearly superior.

Summarising, the strong points of the Web-based run in comparison with the Systran run are its better coverage of proper names and its ability to expand the query. However, sometimes the translation alternatives do hurt retrieval performance, especially when the intended meaning of an unambiguous term in a query is not the most common interpretation. Systran's strong point is its contextual disambiguation, and in particular phrase translation.

FR-EN: Systran versus VLIS. We also looked at some topics that revealed marked differences between the Systran run and the VLIS run. Topic 58 is a clear example where VLIS gives the best results (+0.44) , it correctly translates the key term *euthanasie* by *euthanasia* instead of the non standard translation *euthanasy* by Systran. In most cases however, Systran gives better results, some examples: topic 79 (-1.00), here VLIS fails to translate *Ulysse* into *Ulysses*, the word by word translation strategy also fails for *sonde spatiale*, VLIS translates sonde into *sampler;sound;probe;catheter;gauge;plumb;sink;auger* and spatiale into *spatial;dimensional*. Probably the fact that the query terms *Ulysses* and *space* are missing is more detrimental then the fact that VLIS generates some irrelevant translations for *sonde*, since the correct translation (*probe*) is found. In topic 62 (-0.50) both *Japon* is not found in VLIS and the multi-word unit *tremblement de terre* is not recognised as the French translation of *earthquake*. In topic 66 (-0.50) the crucial proper noun *Lettonie* is not found in VLIS but is successfully translated by Systran. The proper nouns are probably not found in VLIS because in French, country names are usually denoted in combination with a determiner *La France, Le Quèbec,...*, our lexical lookup routine was not aware of this fact. In topic 80 (-0.65) the crucial query term *faim* is translated to *appetite;lust* instead of hunger (Systran). In topic 83 (-0.40), VLIS translates *enchère* by *raise;bid*, whereas Systran gives the contextual better translation *auction*.

Summarising, the Systran-based Babelfish service outperforms the VLIS-based run, because (i) VLIS lacks translations of some proper nouns, (ii) the word-by-word based translation fails for some topics (we currently have not accessed the phrasal translations in VLIS) and (iii) VLIS, which is just a dictionary, has no method for sense disambiguation. Babelfish clearly has a resource for phrase translation: the Babelfish translation in isolation of *enchères* is *bidding*, *ventes aux enchères* gives *auction sales* and *ventes enchères* gives *sales biddings*.

5.5.7.2. *Bilingual runs versus monolingual runs.* Another query-by-query analysis was carried out on a set of queries that exhibited marked differences between the bilingual and monolingual runs. Figure 5.5 shows the absolute performance differences for each topic between the monolingual French run and a run based on the English queries and the 100K QT Web translation model. The average precision of these runs is 0.4233 vs. 0.3878 respectively, but the plot shows that there is a high variability across topics. We selected those topics, where the absolute performance difference was larger than 0.3.

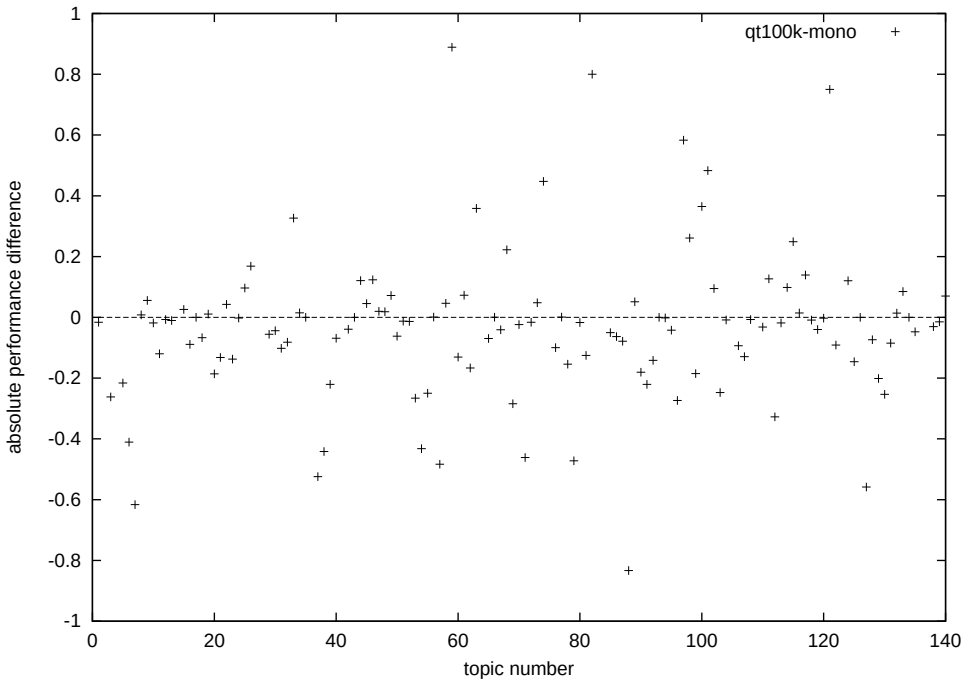

Figure 5.5. Absolute performance differences of the Web-based QT run versus the monolingual run for each topic in the collection

Topics where the monolingual run outperforms the bilingual run.
Topic C005: This topic talks about the European Union. The QT translation of *union* is biased to *syndicat* instead of *union*. (However, the DT translation, based on French to English alignments, is biased towards *union*).
Topic C006: Both *conscientious* and *objector* are not found in the Web-based lexicon.
Topic C007: about *drug use in soccer*. The French translation lacks the salient term *dopage*, which is used in the French version of the topic.
Topic C037: about *sinking ferries*. The French translation lacks *naufrage*, but instead translates root form *sink* with *evier couler lavabo puits*
Topic C038: about repatriation of remains of war casualties. The English topic uses *reburial* instead of *repatriation*, which is missing in the translation model
Topic C054: about *Basketball Final Four*. Here the bilingual version performs worse since *Four* is treated as a stopword and moreover the proper French translation is *demi final*.
Topic C057: about *tainted blood trial*. The bilingual run lacks a translation of *tainted*.
Topic C071: about the relation between eating fruit and vegetables and cancer. In French, one uses the verb *nourrir* in such a context. However, the probabilistic dictionary yields *manger, consommer, nourrir* in decreasing probability. The most common translation of "eating" is out of place in this context, a context-sensitive translation would probably have helped here.

Topic C079: about *space probe/sonde spatiale*. The Web-based dictionary translates *space* with *espace*. Also the bilingual run fails to translate *Ulysses* into *Ulysse*.
Topic C088: about the mad cow disease. Accidentally a literal translation (*maladie de la vache folle*) would suffice, but the bilingual run has the masculin form *fou* instead of *folle*. Also *spongiform* is not found in the translation model, which hurt recall since the translation is *spongiforme*.
Topic 127: The translation model fails to translate *Roldán* into *Roldan*. The latter three topics suggest that a fuzzy matching method could increase performance. We ran an experiment with a fuzzy matching module based on character n-grams (see section 5.4.1). If the retrieval engine does not find a query term in the index, the query term is replaced by the best fuzzy match of the query term in the target collection index vocabulary. Mean average precision increased from 0.3760 to 0.3870, showing that simple orthographic differences can easily be coped by with a fuzzy matching module.

Topics where the bilingual run outperforms the monolingual run.
Topic C033: about cancer. The English topic version uses *cancer* twice, the French version talks about *cancer* and *tumeur*. There are only three relevant documents, none of them contains *tumeur*. The better performance of the bilingual run is thus an artifact due to imprecise (human) topic translation.
Topic C059: about the effects of a computer virus. The French version of the topic mentions *virus ordinateur*. This is correct, but the single relevant French document mentions *virus dans le système informatique*. The bilingual run performs better, since the translation contains both *ordinateur* and *informatique*
Topic C063: The bilingual run performs better because the French query contains a spelling mistake: *Antartique* instead of *Antarctique*, another artifact.
Topic C074: about the inauguration of the tunnel between Britain and France. The French query talks about *Eurotunnel* and *principales personnalités françaises et britanniques*. The English version of the topic is phrased as *Channel Tunnel* and *national representatives of Britain and France*. The bilingual run performs better thanks to several extra relevant terms: *Grande-Bretagne Britannique Angleterre, voie, channel*, the latter (English) term is relevant because a French document talks about the " Channel Tunnel Group" .
Topic C082: about IRA attacks at European airports. There are several reasons why the bilingual run performs better. The English version of the query spells out IRA as Irish Republican Army, attack is translated as attaque, whereas the French original query has *attentat*, finally *terrorist* is translated by *terroriste, terrorisme*, so there is a small contribution from query expansion.
Topic C097: about a referundum on the independence of Moldova. The French version of the topic uses *vote, sondage*. However, relevant French documents use the term *referendum*, not *vote* or *sondage*
Topic C098: about the movies of the Finnish filmmakers Kaurismäki. This is a good example where the bilingual run performs better thanks to query expansion. The translation model brings in the additional terms *tournage, cinéma, cinématographique*,which occur in relevant documents.
Topic C100: about the impact of the Ames espionage case on the US-Russian relations. The French topic phrases the latter as *relations américano-soviétique*, which is correct.

However, the bilingual translations contain *américain, états-unis, russie, russe* which are good expansion terms. Also, the French topic talks about *Aldrich Ames*. Howver, Aldrich does not occurr in the relevant documents, and brings in some irrelevant ones.
Topic C101: about the possibilities of an EU membership for Cyprus. The bilingual run expands EU with *union Européen*, which helps to improve mean average precision.
Topic C121: about the successes of Ayrton Senna. This is a topic with just one relevant document, the English version of the topic is phrased a bit different (*record of sporting achievements* vs. *palmarès*). The better performance of the bilingual run is mostly based on this different phrasing, since the one relevant document does not contain *palmarès* but does contain the Web-based translations: *gagner, réussir, record*.

Concluding, there are many topics with large differences between the performance of the monolingual and bilingual run. On average, the monolingual run performs better. The main explanations for large differences are:

(1) Differences in topic formulation. The CLEF topic creation process is quite complicated. The original topics are developed in several different languages (Womser-Hacker, 2002) and subsequently translated into each topic language. Some topics have thus originally been formulated in French, in English or in other languages. Since those topics have been translated manually, it is hard (or sometimes impossible) to create translated topics, with exactly the same semantic content (in terms of concepts). These translation irregularities explain many of the positive and negative differences between the monolingual French and bilingual English to French results of individual topics. This means that the monolingual baseline, which is often used in CLIR experiments, can only be used as an indicative baseline and that especially comparisons across collections, e.g. 90% of monolingual for English-French against 110% for Chinese-English cannot be made.
(2) Many of the topics where the bilingual run performs worse are due to missing translations or the lack of phrase translations. In a minority of the cases, it is due to lack of context sensitivity (other than phrases), e.g. Topic C071.
(3) The bilingual run performs better than monolingual because of its query expansion capability in several cases.
(4) Perhaps surprisingly, not many of the investigated automatic topic translations suffered from sense ambiguity (see Krovetz & Croft, 1992).

The bilingual run could be improved further by increasing lexical coverage (e.g. using a larger parallel corpus, or adding other lexical resources) and adding context-sensitive (phrase) translation.

5.5.8. Disambiguation and the degree of coordination level matching. One important aspect of translation has not been explicitly covered in the models that were compared in this chapter, the aspect of ambiguity. Hiemstra also performed experiments with several comparable models (see Hiemstra, 2001, chapter 6). In a comparison of probabilistic CLIR models with manual disambiguation, Hiemstra concludes

> "By using the statistical translation model of Information Retrieval, disambiguation is done implicitly during searching."

Hiemstra attributes this fact especially to the fact that queries are structured. After our extensive empirical study, we have a different opinion. We think that the most important reasons that statistical translation performs better than manual disambiguation are:

(1) Sense ambiguity is not a very dominant phenomenon in CLIR experiments (as Hiemstra also noted), although it can be dominant for individual queries.
(2) The power of probabilistic query models, is their ability to handle multiple translations in an effective way (by assigning proper weights), creating a query expansion effect. A human translator (or disambiguator) can never know which translations are used in relevant documents (cf. the previous section for a couple of examples). We have not seen significant differences in performance between the QT (unstructured weighted queries) and DT (structured weighted queries) models.

In order to investigate the query structure aspect a little bit further, we looked at two topics with significant ambiguity. The hypothesis underlying the idea that "structured queries" help to deal with ambiguity in the DT model is that a small value of λ increases the coordination level of the model (see Hiemstra, 2001, section 4.4.4). This means that the model favours documents that contain a translation of each concept.

Let's take the example of a topic with a classical ambiguity: C003, about "Drug policy in Holland". We selected the VLIS runs with all translations (VLIS-all), which translates *drugs* in *remède,hallucinogène, médicament, stopéfier, drogue, narcotique*. We measured the average precision of this topic as a function of the coordination level (the coordination level is inversely related to the smoothing parameter λ). Indeed, we see that the

λ	990	900	700	500	300	100	10
QT-all 100K	0.0287	0.0643	0.08	0.0937	0.1229	0.1285	**0.1306**
DT-all 100K	0.0442	0.1025	0.1811	2092	0.2248	**0.2364**	0.2291

Table 5.16. Influence of smoothing parameter λ on average precision (Topic C003)

coordination helps especially the DT run and since the narcotic sense of *drugs* seems to be related to Holland, coordination helps here.

Another topic with sense ambiguity is C005, about possible new European Union member states. A VLIS-based word-by-word translation translates *union* by *manchon, union, ligue, syndicat*. Again, a low value of λ is optimal here (lower than the globally optimal λ, ensuring a high coordination level. However, the differences between QT and DT are not so marked here. There are several reasons, why this is the case, first of all, *Union* is not as important for the query as *drugs* in the previous example, secondly syndicate often occurs in relevant documents and is thus not such a disturbing translation.

λ	990	900	700	500	300	100	10
QT VLIS	0.0445	0.0607	0.0729	0.0843	0.0945	0.1045	**0.1284**
DT VLIS	0.0576	0.0824	0.0987	0.1048	0.1133	**0.1190**	0.0979

Table 5.17. Influence of smoothing parameter λ on average precision (Topic C005)

We cannot generalize across all topics after just looking at two examples, but we think that indeed structured queries in the sense of the DT model in combination with strong coordination can be effective when the query is relatively short and a key concept has a one or more highly weighted translations, which would bring in irrelevant documents. Since we have tested queries based on the title and description fields, the queries are relatively long. This could be a reason why we do not see marked differences between QT and DT method. Also, it is well known from monolingual IR that the distribution of senses of a word is often skewed (Sanderson & van Rijsbergen, 1999) and that often the most frequent sense is used. In most cases, ambiguity does not really pose a problem if the most probable sense is chosen (using some corpus-based procedure). When the most probable sense according to a corpus is not the intended sense, then there is of course a problem. But of course a query refinement (by substituting a synonym or extension with related terms) step could help here.

We complement this analysis of two queries with an analysis similar to section 5.5.1, where we studied the utility of translation models of different sizes. We plotted the mean average precision as a function of the smoothing parameter for several translation models. Figures 5.6 and 5.7 show the results for Web-based models for 10K and 1M parameters respectively. For 10K parameters, there is not a big performance difference between methods, and the value of the smoothing parameter is not critical. The optimum λ is around 0.1 for most methods, corresponding to a high coordination level. When we add many more "translations", the optimum λ increases a little bit. This is a confirmation that ambiguity is not a big problem, since we would expect a need to increase the coordination level to handle the added ambiguity. We think that the ambiguity that is added by the extra translations is compensated by a query expansion effect. In fact, most of the extra "translations" can better be considered as expansion terms. Since the probability that relevant documents contain all terms from the expanded query decreases, we have to increase smoothing for optimal performance. The optimal smoothing parameter value is thus a trade-off between:

(1) A low value enforces more coordination, favouring documents containing most query concepts
(2) A high value accommodates more expansion terms, since relevant documents not containing a particular expansion term are not extremely discounted.

The non-probabilistic models completely fail to handle the extra "translations", the QT-EQ and NAIVE models perform best with a high level of smoothing, which suppresses the effect of most added translations, but also makes these models less effective, since most coordination is lost. Figures 5.8 and 5.9 show the results of the same experiment based on taking just the main translations or all translations from the VLIS lexical database. The VLIS translations have different characteristics. The set of translations is more constrained and contains less related terms than the Web-based translation sets. Also, the estimated translation probabilities are less precise. The latter aspect is the main reason that there is a smaller difference between the performance of the probabilistic models (QT and DT) and the other models (SYN QT-EQ and NAIVE). Again, the value of λ is not critical.

We conclude that there are no strong indications that structured queries perform better than unstructured queries if we measure average performance on medium sized

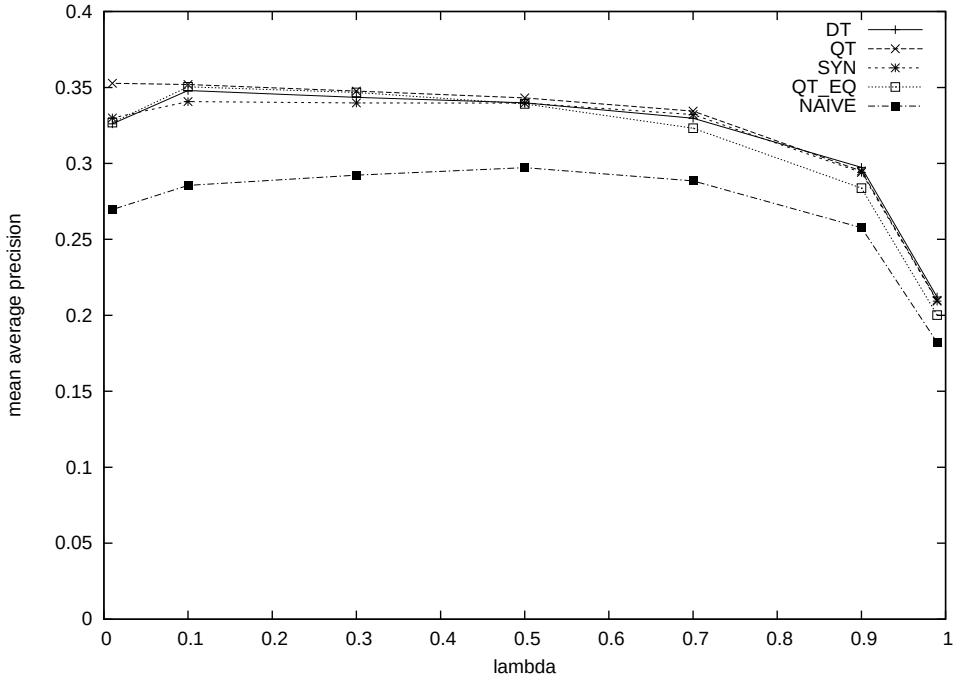

Figure 5.6. Mean average precision as a function of smoothing: 10K model

queries. We think that these queries usually provide enough context for implicit disambiguation, using the coordinative power of smoothed CLIR models. Coordination is also effective for unstructured queries, as long as the query provides enough context. A much more detailed experiment, where the amount of ambiguous concepts, the amount of context and the number of expansion terms is controlled or quantified is necessary to investigate these effects in more detail. We think that CLIR performance can still be considerably improved, by optimizing the CLIR model for different classes of queries. Starting point could be the work of Cronen-Townsend et al. (2002), who propose the clarity score as a measure for query ambiguity. The clarity score correlates well with query effectiveness.

5.6. Conclusions

We have studied the problem of cross-lingual query-document matching in an integrated probabilistic framework. Our goal was to define and evaluate combinations of probabilistic generative models and translation resources that are at least as effective as the combination of MT and monolingual IR. In particular we wanted to validate different embeddings of word-by-word translation models in a language modeling based IR framework. Previous work (Kraaij, 2002; Kraaij & Pohlmann, 2001) had given indications that there were potential interactions between the type of resources (dictionaries, parallel corpora, bilingual term-lists) and the types of models (quasi Boolean, statistical translation,

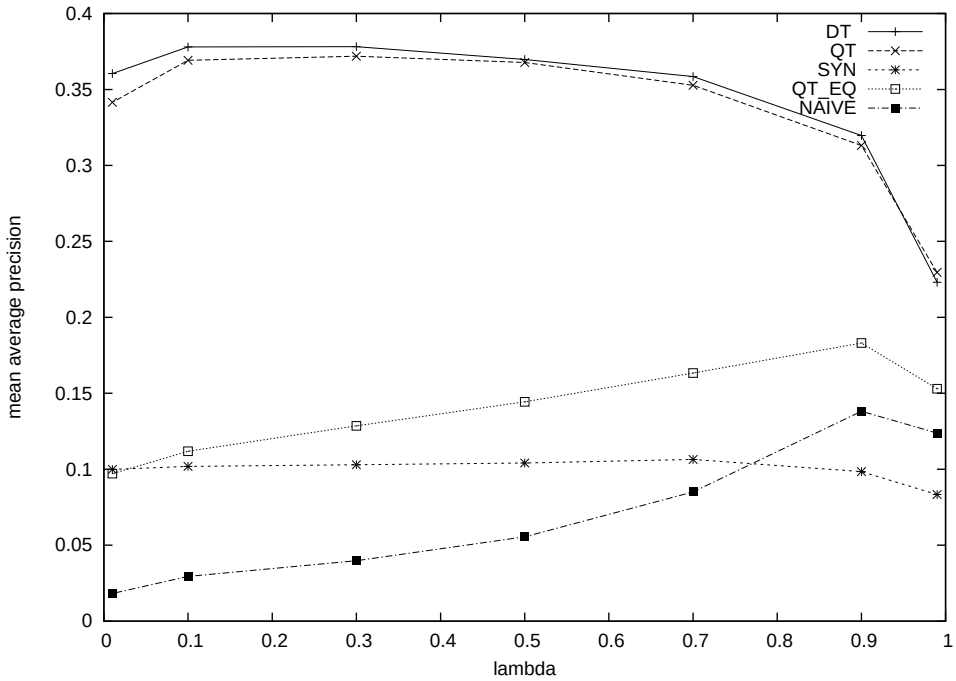

Figure 5.7. Mean average precision as a function of smoothing: 1M model

translations as synonyms). This study has explored the interactions between models and resource types by a set of carefully designed experiments. Although word-by-word translation is a quite naive approach to the problem of translation, it nevertheless yielded good results for CLIR, reaching on average almost the same level as a monolingual run, but surpassing monolingual runs in many individual cases.

We have shown that there are several possibilities to integrate word-by-word translation in a retrieval model, namely by mapping a language model of the query into the target language (QT), mapping a language model of the document into the source language (DT) or mapping both into a third language (QT+DT). The well-known Pirkola method for CLIR can be regarded as a non-weighted instantiation of the DT method. In order to perform a systematic study regarding the possible interaction between the theoretical models and the different types of resources, we also evaluated several simpler models i.e. a model with equal translation probabilities, a model based on naive replacement and a model taking just the most probable translation for each term.

All these models require simple translation models: a weighted translation matrix representing the probability that a certain word in the target language is the translation of a word in the source language or vice versa. Parallel corpora are the ideal resource to build these simple statistical transfer dictionaries, since the corpus statistics provide a good resource for estimation. Since parallel dictionaries are not always easy to acquire, we have investigated whether the Web can be used as a resource. We also converted a lexical database into a statistical transfer dictionary using simple heuristics in an attempt

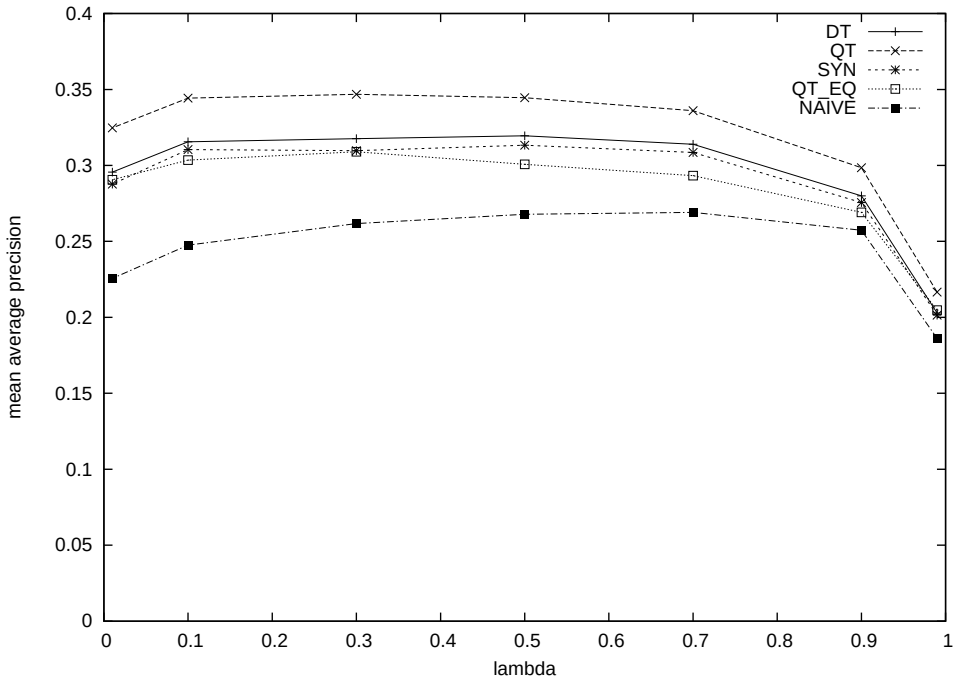

Figure 5.8. Mean average precision as a function of smoothing: VLIS main translations

to produce reasonable translation probability estimates. The two types of resources have very different characteristics. The Web-based models have more accurate probability estimates, but their coverage critically depends on the number and diversity of bilingual sites crawled. Since we used simple but robust sentence and word alignment models, the Web based models will always contain a certain amount of noise, spurious translations. Pruning these spurious translations has proved to be an essential step to make these translation models usable for CLIR. Especially the DT model benefits from pruning, since it is highly sensitive to spurious translations.

We evaluated the different CLIR models by measuring retrieval performance on a test collection consisting of approximately 125 topics and three document collections from American, French and Italian newspapers. The probabilistic integrated CLIR models yielded significantly better performance than the runs that were based on "external translation" by the commercial Systran translation service and thus showed their great potential despite their simplicity. Results were especially good with the Web-based translation models. Probabilistic CLIR models based on Web-based word-by-word translation systematically outperformed the unweighted translations-as-synonyms CLIR method. The good results can be attributed to two aspects: (i) better translation weights, (ii) query expansion using the parallel corpus. We have not observed a systematic performance difference between the different directions of the probabilistic CLIR models. Performance differences seem merely due to quality differences in the translation models. We cannot confirm that the fact that the DT model uses structured queries is a

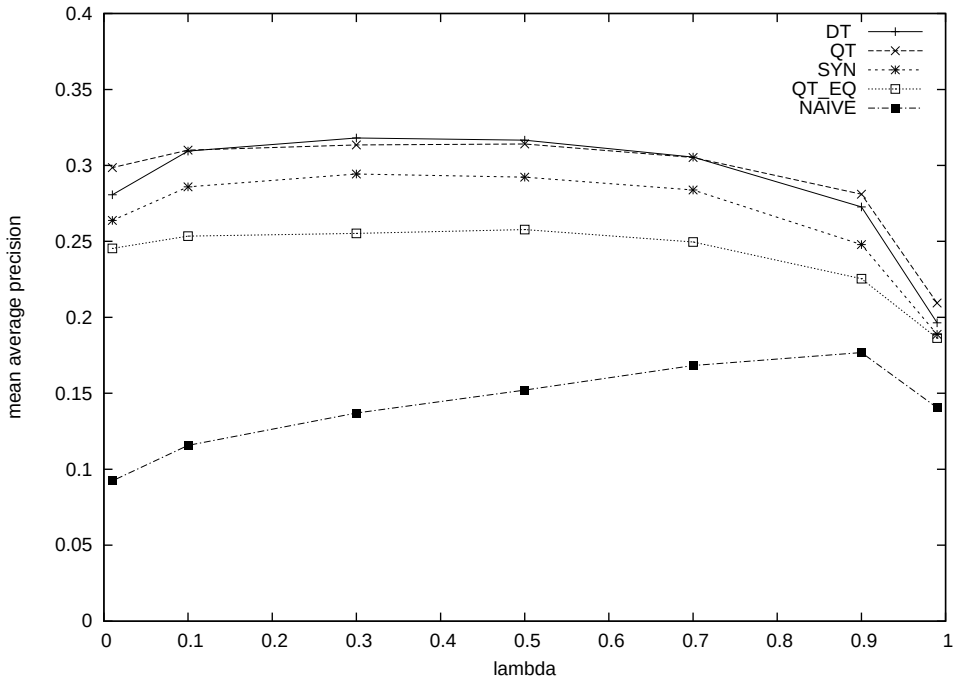

Figure 5.9. Mean average precision as a function of smoothing: VLIS all translations

determining factor for an effective CLIR model. Proper relative weighting of translation alternatives seems much more important. When we have to deal with many translation alternatives, the weighted QT and DT models really outperform the non-weighted SYN and NAIVE models and have thus shown to be robust. The importance of weighting is also demonstrated by the fact that choosing the most probable translation is more effective than weighting all translations equal (for the 100K models).

We also investigated the interaction of the amount of coordination enforced (coordination is inversely correlated with the level of smoothing) and the performance of the QT and DT models. Indeed a high level of coordination seems to be required for the best performance on queries with highly ambiguous terms. But on average, we did not find significant differences between structured and unstructured approaches for medium length queries. There is something to say for the unstructured queries as well, since it seems to be a more natural model to accommodate expansion terms, which we have found come naturally with a translation model trained on parallel corpora.

This performance difference is reversed for the VLIS-based translation models, which lack a good resource for proper probability estimation. In fact, choosing just the most probable Web-based translation results in a performance comparable to choosing all (weighted) translations from the VLIS database. This is mostly due to the poor probability estimates of the VLIS-based dictionary. Using POS information to constrain lexical lookup resulted in a very small improvement in mean average precision. Part-of-speech

ambiguity is not very frequent and sometimes translations of homonyms with a different POS category result in effective query expansion.

We also extended the CLIR model with transitive translation and showed that acceptable performance can be reached (63%-83% of a monolingual setting). Effectiveness seems to be largely determined by the coverage of the smallest translation model. The same method can be used to improve monolingual expansion, which proves that translation alternatives can be effectively used to enhance recall.

A simple way to increase CLIR effectiveness is to increase lexical coverage. We showed that by combining resources in a straightforward fashion, performance could be increased up to a level of 99% of monolingual effectiveness. We think that there is still room for improvement, since we were able to improve retrieval effectiveness of a monolingual run, using bilingual dictionaries or parallel corpora.

We think that there is still room for substantial improvement of CLIR systems. It is a matter of finding good resources and combining them in a robust way. The most important missing component in our CLIR system is the translation of multi-word-units (including phrases). This is an important area for further potential performance gains. Another important technique that could increase retrieval effectiveness is pseudo feedback (pre- or post translation). We have deliberately refrained from using this technique to keep the experiments well controlled. However, the border between deriving a translation model from a parallel corpus and expansion via a parallel corpus is quite thin. It might be that an integrated model for translation and expansion could be quite effective, decreasing the need for more complex translation models.

CHAPTER 6

Stemming methods and their integration in IR models

In this chapter, we will review and compare several ways to deal with morphological variation in an ad hoc monolingual retrieval task. We will limit our study to Dutch, though many of the aspects of dealing with a language which has a more complex morphology than English are also valid for other European languages (Hollink et al., 2003). We will also revisit some of our early research results on stemming and decompounding for Dutch (Kraaij & Pohlmann, 1996b; Pohlmann & Kraaij, 1997b) in the context of a generative probabilistic framework. Cf. section 3.3.1 for an introduction into morphological normalization and its application in IR.

Traditionally, stemming has been approached as a preprocessing step to be applied before indexing and retrieval, motivated by the observation that there is often a mismatch between terms used in the query and the related terms in relevant documents. One of the sources of mismatches is morphological variation, the standard cure is to normalize document and query terms e.g. by removing suffixes, based on the intuition that morphological variants are instantiations of the same semantic concept. A corresponding IR model will be based on a feature space of equivalence classes instead of a feature space of wordforms. Each equivalence class consists of the wordforms that share the same morphological base form. One could argue that this is a heuristic approach, just like the removal of stop words, because these techniques are not part of the retrieval model itself. The generative probabilistic modelling approach to IR seems to have the potential to accommodate morphological normalization as a part of the IR model (Ponte, 2001) in a more motivated fashion.

This chapter consists of two main parts, preceded by an introducing section 6.1, which describes some baseline experiments without morphological normalization. In the first part (section 6.2), we will compare suffix stripping, full morphological analysis and fuzzy matching. In the second part (section 6.3), we will discuss and reinterpret the early experiments concerning morphological normalization based on (naive) query expansion Kraaij & Pohlmann (1996b) and evaluate alternative methods to combine stemming and term weighting in a wordform feature space based on a language modeling framework.

6.1. Baseline experiments

The experiments described in part I of this chapter were carried out with two different retrieval engines and various weighting algorithms. In order to set a baseline, we will

describe some experiments with bare versions of these systems (section 6.1.2). They give a rough indication of the quality of the different term weighting algorithms, but in addition these experiments give an indication of the range of performance gains that one can get by improving term weighting models. Since pseudo relevance-feedback is known as a very effective query expansion method and one way to use morphological equivalence classes is to use them for query expansion, we will run some experiments with this method as well (section 6.1.3). The complete set of baseline results can serve as a contrast to the experiments that are discussed in the sections about morphological normalization (section 6.2).

6.1.1. Description of search engines. In this section we will describe the two IR engines[1] used for the experiments with monolingual retrieval for Dutch.

TRU engine. The early series of experiments, which has been published in (Kraaij & Pohlmann, 1996b), (Pohlmann & Kraaij, 1997a) and (Kraaij & Pohlmann, 1998) was carried out with the TRU (Text Retrieval Utility) vector space engine developed at Philips Laboratories (Aalbersberg et al., 1991). This engine was implemented in the spirit of Salton's SMART system, but heavily optimized for usage in limited memory applications and for work with dynamic data. These conditions motivated a choice for a *nnc* term weighting scheme (cf. Appendix A) for both documents and queries (an *idf* component would require a recalculation of all term weights after a database update). The *idf* component was introduced as a separate external factor:

$$\mathrm{RSV}(\vec{q},\vec{d_k}) = \sum_{i=1}^{T_q} \frac{tf_{q,i}tf_{k,i}}{\sqrt{\sum_{i=1}^{T_q} tf_{q,i}^2}\cdot\sqrt{\sum_{i=1}^{T_{d_k}} tf_{k,i}^2}} \cdot idf_i \tag{80}$$

The engine had only little possibilities for tuning the term weighting, because the source code was not available. Only the rather ad hoc nnc.nnc.idf based scheme (cf. appendix A) could be used. The only two options we had for experiments, were to modify the stemming algorithm and to pre-process queries and/or documents in order to influence term weighting.

TNO engine. A lot of the TRU experiments were run again with the retrieval engine we developed at TNO TPD. This engine had the advantage of complete access to the source code. The object oriented implementation made it relatively easy to experiment with different variants of techniques, simply by setting parameters or replacing modules. We experimented with several vector space models and probabilistic models. The only limitation for the engine was that term weighting schemes had to be rewritten in a presence-only ranking scheme, i.e. with documents ranked only on statistical information about the terms that they share with the query and global data like document length, which can be pre-computed off-line. The TNO engine has been used in a large number of benchmark tests: TREC-[6-10], CLEF200[0-2] and has proved to be robust, efficient and flexible. The experiments with Okapi and LM-based term weighting that are reported in this thesis (chapters 5 and 6) were all conducted with this engine.

[1] We use the word "engine" for the actual implementation of a retrieval model.

6.1.2. Results of baseline experiments. All experiments described in this chapter were carried out on a collection developed by Kraaij and Pohlmann (the UPLIFT test collection). Details about the development of the collection can be found in Appendix C. The collection consists of 59608 documents from Dutch regional newspapers (year 1994) and 66 queries.

Table 6.1 and figure 6.1.2 show the performance of several baseline systems. The first system (with label nnc.nnc.idf) is the TRU system. The other runs are produced by the TNO engine, using different term weighting schemes. The second baseline: BM25 refers to the Cornell version of the Okapi system (cf. formula 30). The third and fourth baseline runs are based on an LM based IR model. The third baseline is an implementation of the original model of Hiemstra using document frequencies for the estimation of the background model term probabilities: $P(\tau_i|C) = df(\tau_i) / \sum_i df(\tau_i)$ (cf. formula 35). The fourth baseline is the same LM based model, but using collection frequencies for the background model term probabilities: $P(\tau_i|C) = \sum_D tf_i / \sum_i \sum_D tf_i$

All runs were based on a 1326 word stop list (cf. section 3.4), which is a mixture of high frequency terms and closed classes (prepositions, pronouns etc). All index terms were converted to lower case, no stemming was applied. A comparison of the four

version	ap5_15	% change	map	% change	R-recall	% change
nnc.nnc.idf	0.438		0.284		0.310	
vBM25	0.481	+ 10	0.322	+ 13	0.356	+ 15
vLM-plain-df	0.471	+ 7	0.318	+ 12	0.348	+ 12
vLM-plain-cf	0.453	+ 3	0.301	+ 6	0.335	+ 8

Table 6.1. Comparison of basic systems on UPLIFT test collection. AP5_15 is the averaged high precision, map is the mean average precision and R-recall is the recall measured at R documents. (cf. section 4.3)

baseline systems shows that the runs based on the TNO engine (i.e. the more recent IR models) perform somewhat better. The BM25 run and the vLM-plain-df run score significantly better than the TRU run (5%level). Interesting is the fact that vLM-df is significantly better than vLM-cf at the 1% significance level. The document frequency based estimate is apparently more robust, probably since document frequencies are smooth out highly skewed term frequencies across documents. The collection frequency based estimator is much more sensitive to outlier documents.

6.1.3. Adding pseudo relevance-feedback. In chapter 2 , section 3.1 we discussed several techniques for relevance feedback. Relevance feedback procedures use explicit user feedback on relevance of documents to improve the term weighting function. A good example is the classical Robertson-Sparck-Jones formula. In some special cases, relevance feedback can also improve retrieval effectiveness without user feedback, by assuming that the top retrieved documents are relevant. This assumption is valid when: (i) the document contains at least a few (5-10) relevant documents for a query and (ii) the IR system can retrieve some of those in top position (this is for a large part dependent on the query formulation). Assumption (i) is often met for IR tasks at recent IR benchmarking conferences like TREC. In the topic selection process, the organizers select topics

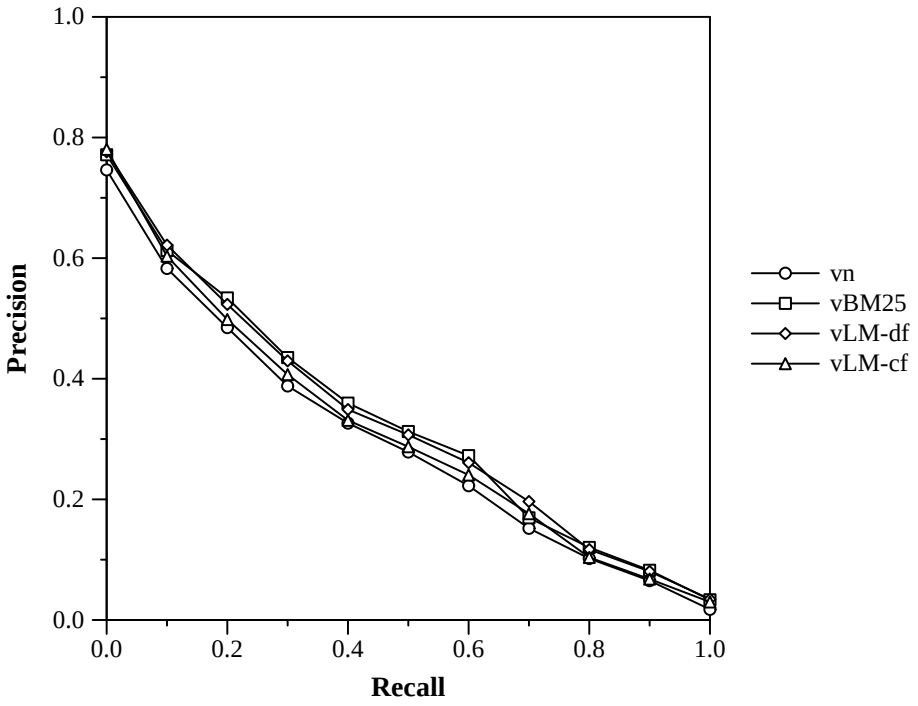

Figure 6.1. precision-recall graph of the performance of our baseline systems on the UPLIFT test collection

that have at least a certain minimum number of relevant documents in the database. The second assumption is usually met by state-of-the-art IR algorithms. On average, automatic relevance feedback procedures are reported to improve retrieval effectiveness by 10-25%, depending on the test collection, the quality of the baseline and the quality of the feedback algorithm. We think that these results cannot always be generalized since the improvement is partly due to exploiting the prior knowledge that queries are well formulated and that the collection contains relevant documents. An example where one of these conditions does not hold is the CLEF multilingual task. In this case, the topic creation process implies that only one or more sub-collections contain relevant documents for a query, consequently some collections do not contain relevant documents or just one or two. Automatic relevance feedback methods may deteriorate results in these cases.

Although relevance feedback is not the focus of our research, we have implemented two automatic relevance feedback procedures, in order to produce some reference data, to which we can compare the linguistic methods presented in the remainder of the chapter. Table 6.2 presents results of a baseline system based on the Okapi BM25 formula and a feedback run, based on blind relevance feedback (3.1.4), using 200 terms from the top three documents.

version	ap5_15	% change	map	% change	R-recall	% change
vBM25	0.481		0.322		0.356	
vBM25-brf	0.524	+ 9	0.386	+ 20	0.409	+ 15

Table 6.2. Pseudo Relevance Feedback

6.2. Comparing different approaches to morphological normalization

In this section we will compare several alternative ways to achieve morphological normalization for monolingual Dutch IR. The standard method to apply morphological normalization for IR is to treat all morphological variants of a lemma as instances of the same equivalence class, i.e. by substituting them with a single string, usually (but not necessarily) a stem. This process is also called *conflation*. Conflation can be performed either at indexing or retrieval time. The latter method is also called "on-line stemming" and is described in more detail in section 6.3.1.

Our basic research question is whether a full fledged morphological analysis will give a significant gain in retrieval performance with respect to methods with lower levels of linguistic sophistication: a "Porter" like stemmer, which consists of a rule-set of limited coverage or stemming as fuzzy matching which is a more heuristic approach. Morphological normalization is thus evaluated in the context of an IR experiment. Since the evaluation is based on just a small number of topics (66) it can never serve as an evaluation of the accuracy of morphological normalization per se, but that is not our goal. A more principled evaluation of the accuracy of the Dutch stemmer of Kraaij & Pohlmann has been presented in Kraaij & Pohlmann (1995). Such an analysis is especially helpful for the development of stemming rules.

A secondary research question concerns nature of stemming. Stemming is usually considered a recall-enhancing device, but to what extent do the experiments confirm this intuition?

6.2.1. Conflation variants: full morphological analysis. We developed two linguistic stemmers (inflectional and derivational) using a computer readable dictionary, the CELEX lexical database (Baayen et al., 1993). Using CELEX, two separate files were created which relate stems to their inflectional and derivational forms respectively. The inflectional stemmer is quite conservative since it only groups inflectional forms. The derivational stemmer is much "stronger" since it groups derivational forms (including their inflectional variants) with the morphological base form: e.g. *kunstmatig* → *kunst*. To avoid unnecessary overhead, not all possible forms were included in these files but only those forms which actually occurred in our test collection. In cases of lexical ambiguity, when a particular string can be related to two different stems (e.g. *kantelen* can either be related to the noun stem *kanteel* ('battlement') or the verb stem *kantelen* ('to turn over')) we simply selected the most common interpretation based on frequency information provided in the CELEX database. The files were used to implement on-line stemming, i.e. we applied the method described in section 6.3.1.4.

In table 6.3 the results of the derivational stemmer (vTNOderivc_gr) and the inflectional stemmer (vTNOinflc_gr) are compared with a baseline of no stemming. All runs

were based on the Okapi probabilistic model.[2] It is clear that dictionary based stemming in general is effective for Dutch. Both the inflectional and the derivational stemmer improve on the baseline for all three evaluation measures (at significance levels of 0.05 and 0.01 respectively as measured by a sign-test). Apparently, Dutch morphology is complex enough for stemming to have a beneficial effect on retrieval performance.

When comparing the results of the inflectional and derivational stemmer we find that, on average, the derivational stemmer is slightly better than the inflectional stemmer, except at high precision (ap5-15) where the effect is reversed. The performance of the dictionary stemmer might be improved by restricting conflation to morphological variants that have the same sense, for example by measuring the distance between context vectors (Jing & Tzoukermann, 1999) or constraining stemming to forms that co-occur frequently in a small text window (Croft & Xu, 1995).

version	ap5_15	% change	map	% change	R-recall	% change
vTNO-baseline	0.481		0.322		0.356	
vTNOderivc_gr	0.512	+ 6	0.372	+ 16	0.392	+ 10
vTNOinflc_gr	0.519	+ 8	0.366	+ 14	0.383	+ 7

Table 6.3. results CELEX experiment

6.2.2. "Porter" for Dutch. In 1980, Porter published a stemming algorithm based on suffix stripping that became a reference implementation for many IR researchers (Porter, 1980). The algorithm consists of several classes of rules, which are interpreted class by class. If a rule in a certain class matches the input string, it is executed, which means that a suffix is removed or changed. In Kraaij & Pohlmann (1994), the development of a Dutch version of this algorithm is reported. The algorithmic structure is based on the English original and consists of 98 rules which fully cover Dutch regular inflectional morphology and partly cover derivational morphology. In addition to suffixes, the algorithm also removes some pre- and infixes, in order to stem past participles. For a more detailed description of this stemming algorithm can be found in Kraaij & Pohlmann (1994)[3]. Porter has also developed stemmers for several other European languages including Dutch, originally for Muscat Ltd. Eventually some of the source code of Muscat became open source[4], currently the original (Open) Muscat stemmers are available under the name Snowball stemmers and have been used by many IR researchers working on European languages. The results of the experiment with the Kraaij&Pohlmann stemmer

version	ap5_15	% change	map	% change	R-recall	% change
vTNO-baseline	0.481		0.322		0.356	
vTNOporter2_gr	0.519	+ 8	0.362	+ 13	0.388	+ 9

Table 6.4. results Porter experiment

[2] the runs vBM25 and vTNO-baseline are identical.
[3] The source code is available at `http://www-uilots.let.uu.nl/uplift/`
[4] (see `http://www.xapian.org/history.php`)

are presented in table 6.4. The Dutch Porter algorithm proves quite effective on the test collection. It improves upon the baseline for all three evaluation measures. Sign-tests showed that differences were significant at the 0.01 level. We therefore conclude that Porter stemming is a viable option to improve retrieval performance for Dutch texts. A precision-recall graph of the results of the Porter stemmer in comparison with the results of the two dictionary-based stemmers is presented in figure 6.5. The figure shows that

version	ap5_15	% change	map	% change	R-recall	% change
vTNO-baseline	0.481		0.322		0.356	
vTNOderivc_gr	0.512	+ 6	0.372	+ 16	0.392	+ 10
vTNOinflc_gr	0.519	+ 8	0.366	+ 14	0.383	+ 7
vTNOporter2_gr	0.519	+ 8	0.362	+ 13	0.388	+ 9

Table 6.5. Porter vs. CELEX

the performance of the Porter stemmer is completely comparable to the performance of the two dictionary-based stemmers. In fact, the difference is not statistically significant for any of the three evaluation measures[5]. So, although the Porter stemming algorithm is much simpler and theoretically much more error-prone than the dictionary-based algorithms, in practice it seems to be just as effective. We conclude that it may be preferable to use the Porter algorithm instead of one of the dictionary-based algorithms, at least in a monolingual Dutch retrieval setting. A suffix removal approach is not optimal in a CLIR setting using translation dictionaries, since there is often a mismatch between the word stem and the lemma of a word. dictionary entries could be stemmed as well, but this results in a sub-optimal use of the dictionary (cf. section 5.1.3).

6.2.3. Conflation variants: Fuzzy Matching. In a third comparison experiment we tested whether we could implement term conflation with even less linguistic information. We evaluated the use of *fuzzy matching* for variant term matching.

From a conceptual point of view, the idea to use fuzzy matching for stemming is quite simple. We started this chapter with the assumption that morphological variants are instantiations of the same semantic concept. In this section we investigate the performance of a system based on a weaker assumption: words with a *similar orthographic form* have a high probability to be related to the same semantic concept. In principle a conflation procedure based on this assumption does not require any linguistic knowledge. Conflation could be realized by using a metric which measures the orthographic similarity of words. Suitable techniques can be found in the field of approximate string matching (cf. section 3.2) . In the following sections we will describe the tools we used for conflation based on approximate string matching and the experimental results.

Fuzzy conflation architecture. Fuzzy conflation is in some ways different from the previous conflation techniques. For stemming and lemmatisation, the universe of indexing features V is reduced to a new universe V' which is based on equivalence classes, characterised by a common root form. For every indexing feature in V there exists a mapping

[5] cf. chapter 4 section 4.4 for a more elaborate discussion of statistical tests.

into an indexing feature in V'. Stemming and lemmatisation are thus functions. However, fuzzy conflation is not a function, the equivalence classes are defined by clusters of words which fulfil some constraint on a similarity measure with a certain keyword. An example of such a constraint is: *the equivalence class for 'walks' is formed by all words which share at least 2 trigrams with walk.* If we define an equivalence class for the word 'talks' in the same way, it is evident that equivalence classes can overlap, which will deteriorate precision. Tightening the constraints for membership of an equivalence class on the other hand will decrease recall. The fact that equivalence classes can overlap is not desirable for indexing, since it increases dependency between indexing features. On the other hand, it is well known that some term dependence is not a problem for the effectiveness of IR models.

For our experiments, we decided to ignore potential problems due to increased term dependence. We generated the equivalence classes on the fly, as a form of fuzzy query expansion, e.g. for each query term we generated an equivalence class based on the indexing dictionary and a similarity metric. Because computing the similarity of a query term with each word in the main indexing dictionary (of the vector space index) would be too inefficient, we used ISM (cf. section 3.2) to build a secondary trigram index on the indexing dictionary, in order to be able to generate on-line equivalence classes in an efficient manner. The trigrams were extracted from words that were padded with one leading and one trailing blank. Trigrams did not cross word boundaries by default and no positional information was kept. The similarity metric of ISM is based on the distributional properties of the trigrams, but the algorithm is also tuned for an optimal performance. Because ISM's primary aim is retrieval of strings which are similar to a query and not similarity per se, ISM could not be used for conflation right away. We post-processed the ISM output by scoring on an external similarity measure, namely the Levenshtein edit distance metric (cf. section 3.2.1) . For our fuzzy expansion experiments, we used ISM as a fast lookup and first similarity step, Levenshtein was used to select only those wordforms within a certain edit distance from the original query term.

In order to use ISM and the Levenshtein metric as a term conflation tool, the following steps were carried out:

(1) *Index the UPLIFT document collection on word forms (off line).* This was done by using the default TNO vector space engine with stemming disabled.
(2) *Extract the indexing dictionary from the index (off line).* The indexing dictionary can be extracted easily because it is stored as a separate file on disk. The indexing dictionary is composed of full word forms, because stemming was disabled during indexing.
(3) *Build a fuzzy index on the word form list(off line).* The TNO ISM tool was used to build an n-gram index i.e. each term in this dictionary is indexed by overlapping trigrams as indexing features.
(4) *For each query term:*
    (a) *Select the most similar terms in the indexing dictionary using the fuzzy index (on-line).* The query term was decomposed into a set of overlapping trigrams. These trigrams were ranked using the ISM algorithm (de Heer, 1979), which is (a.o.) based on the distributional properties of the trigrams.

Note that the usage of ISM for ranking variants is not critical for this particular application. In more recent conflation experiments for CLEF Hiemstra et al. (2001b), we applied standard *tf.idf* like trigram weighting methods, which proved equally effective.

(b) *Optional (1): Exclude words that exceed a pre-specified edit-distance (online).* We experimented with an extra similarity post-processing filter, because the default ISM algorithm does not penalize substring matches.

(c) *Optional (2): Exclude words that start with a different character.* A second fine-tuning step constrains expansion terms to words that start with the same character.

(d) *Use the resulting expansion set as a conflation class using the grouping operator.* A *grouping operator* ensures that the words in a fuzzy expansion will be treated like an equivalence class which is formed during stemming and has semantics similar to INQUERY's SYN operator. This means that each occurrence of one of the terms in a document adds to the term frequency of the class. The document frequency of an equivalence class is defined as all documents that contain at least one term of the conflation class. The document frequency of the class is thus always equal to or larger than the collection frequency of any of the class members. The collection frequency is defined as the sum of the individual collection frequencies. We will discuss the grouping operator in more detail in section 6.3.1.4

Experiments. We performed a series of experiments to test the feasibility of conflation based on fuzzy matching. All tests are based on the TNO engine with Okapi weighting. All experiments are based on query expansion using the grouping operator. The first

version	ap5_15	% change		map	% change		R-recall	% change	
vTNO-baseline	0.481			0.322			0.356		
vISM90	0.292	−	39	0.200	−	38	0.240	−	33
vISM01	0.494	+	3	0.336	+	4	0.362	+	2
vISM02	0.514	+	7	0.356	+	11	0.384	+	8
vISM02-nofirst	0.496	+	3	0.340	+	6	0.373	+	5
vISM03	0.507	+	5	0.355	+	10	0.383	+	8
vISM04	0.493	+	2	0.345	+	7	0.374	+	5

Table 6.6. Fuzzy Stemming

experiment of this series used fuzzy matching without any further restrictions (vISM90). We configured ISM to limit query expansion to terms with a similarity of at least 90%. Results were rather disappointing. Inspection of the term expansions revealed that this was most probably due to ISM's insensitivity to differences in string length (if the query term is a substring of another term, their similarity is about 90-95%, note that the similarity measure in ISM is not symmetric). We modified ISM to use the Levenshtein edit-distance as a post filtering step, we also found that we could improve upon these results when restricting expansions to words starting with the same character. The runs vISMn where n is the maximal edit distance are runs with all restrictions in place. Omitting the first

character restriction (vISM02-nofirst) results in a noticeable drop in performance. The optimum edit-distance turned out to be 2. A larger edit-distance often brings in unrelated terms. For the edit-distance of 2, we compared 'standard grouping' with naive expansion (cf. section 6.3.1), the difference was quite significant (0.3562 vs. 0.2737). Concluding we can say that the combination of fuzzy matching, edit distance and the first character heuristic performs quite well. This run (vISM02) performed significantly better than the baseline at the 0.01 level.

Finally, for the Porter and CELEX based methods with the fuzzy matching based conflation, differences are quite small, which is a striking result. But of course these tools could be seen as an implementation of some very basic heuristics about morphology: morphological related words (in Dutch) most often start with the same letter and most often differ only a little bit in orthographic form. Apparently, the fact that conjugates share some common base form is enough to build effective conflation techniques. Section 3.2 presents some pointers to other (more recent) work on the application of n-grams for IR, which also shows that n-grams can be an effective means to deal with morphological variation. Our approach is different in the sense that we constructed a cascaded index: we indexed the documents by full wordforms and indexed the wordforms by n-grams. Other researchers indexed the documents directly by the n-grams. They usually found that n-grams alone perform worse than wordforms, but can be affectively applied in a combination approach (Mayfield & McNamee, 1999; Hollink et al., 2003). We also tried a combination of Porter and fuzzy matching: fuzzy matching + edit

version	ap5_15	% change	map	% change	R-recall	% change
vTNO-baseline	0.481		0.322		0.356	
vTNOporter2_gr	0.519	+ 8	0.362	+ 13	0.388	+ 9
vp2ISM01	0.510	+ 6	0.366	+ 14	0.390	+ 9
vp2ISM02	0.511	+ 6	0.370	+ 15	0.392	+ 10
vp2ISM03	0.512	+ 6	0.369	+ 15	0.391	+ 10

Table 6.7. ISM + Porter

distance on a Porter-stemmed index. The combination vp2ISM02 gave a slight improvement for the average precision but also a slightly worse high precision. This approach does not seem to be effective in terms of resources (a fuzzy conflation run is slower since it involves lookup and online conflation through expansion) vs. results.

Conclusions. Conflation based on fuzzy matching is nearly as effective as Porter stemming or CELEX lemmatisation. An effective fuzzy matching algorithm can be constructed by combining a fast look-up pre-selection procedure based on trigram matching with the Levenshtein based edit-distance. An essential component for emplorying fuzzy based conflation methods in an IR engine is the grouping technique, which reduces term dependencies. Naive query expansion introduces a lot of query term dependencies and therefore performs disappointingly. The fuzzy conflation technique is promising as a simple language independent stemming technique. No time consuming coding of morphological rules is necessary.

6.2.4. Compound analysis. We also performed several experiments with compound splitting for Dutch. Compounding is an important phenomenon for Dutch. Approximately 37% of the wordforms in the UPLIFT document collection which were not included in the CELEX dictionary are compounds (Kraaij & Pohlmann, 1996b). In Dutch, nominal compounds are generally formed by concatenating two (or more) words to create a single orthographic word, e.g. *fiets* ('bicycle') + *wiel* ('wheel') → *fietswiel*. As compounding is a very productive process in Dutch, every dictionary is necessarily incomplete in this respect. To handle this problem, some stemmer versions were extended with a compound analyser, the 'word splitter' developed by Vosse for the CORRie (grammar checker) project (Vosse, 1994). The word splitter tries to split a compound into its components (stems) on the basis of word combination rules for Dutch and a lexicon. If the splitter is unsuccessful, the word is left unchanged. The accuracy of the compound splitter was evaluated on a random sample of approximately 1,000 compounds not included in the CELEX dictionary[6]:

5%	no analysis
3%	incorrect analysis
92%	correct analysis

Table 6.8. Evaluation of the accuracy of the Vosse compound splitter

The embedding of compound splitting in IR is not a trivial problem. It is not always clear whether a compound should be split, e.g. we do not want to split the compound "hoofdstuk" (chapter) since the compound parts "hoofd" (head/main) and "stuk"(piece) are hardly related to the meaning of the compound as a whole and thus might introduce a lot of unwanted matches. There are several possibilities:

Expansion: add parts to query: In a pure on-line setting, where conflation takes place at run-time, we can split all compounds in the query and subsequently add all the compound parts to the query.

Expansion: split and generate: In a pure on-line setting, where conflation takes place at run-time, we can split all compounds in the query and subsequently add all compounds to the query that occur in the document collection and whose stems occur in the query. This process will help to match the query "vervuiling van water" with a document containing "watervervuiling". Since the procedure is unconstrained, this sometimes leads to expansion with some unrelated terms. This option and the previous one are discussed in more detail in Kraaij & Pohlmann (1996b). Although, this study contained only some very preliminary results on compound analysis, it indicated the importance of compound splitting to achieve a good recall.

Syntactic Analysis: add heads or all parts: We experimented with shallow NP parsing in order to constrain matching between compounds and other noun-noun constructions (e.g. PP-modification) in a more principled way. The idea was that compounds are a form of noun phrases and that we could have a precise match through normalization of noun phrases by reducing them to head-modifier

[6]Some frequent compounds are included in the CELEX dictionary.

pairs. We found that mean average precision could be enhanced by 12% and R-recall by 22% provided all parts of the identified phrases were also added to the index. Slightly better results were achieved by adding head-modifier pairs of noun-noun and/or adjective-noun constructions to the index Experiments were based on the TRU retrieval system and are described in detail in Pohlmann & Kraaij (1997a). Adding all compound parts to the index might create a problem though, since the relative weight of compound terms w.r.t. other terms is artificially inflated.

Replace by (stemmed) parts: Replace compounds in query and document by compound parts. This is essentially a control experiment to investigate the effect of the weight inflation of compound terms. We found that replacing a compound by its part is less effective then adding the parts and the original (stemmed) compound form the index (Kraaij & Pohlmann, 1998).

6.2.5. Discussion.

Comparison of stemming methods. We have compared three stemming techniques for Dutch that did not involve compound splitting: suffix stripping, dictionary based stemming and fuzzy matching. The principal representatives of these methods are shown in figure 6.2. The different strategies have been presented in a decreasing order of linguistic motivation. The pure morphology based stemmers based on the CELEX morphological database performed best. But the effectiveness of the much simpler Porter algorithm is quite similar, differences are actually not statistically significant. The suffix stripper encodes a sufficient amount of morphological knowledge to be as effective. Even the fuzzy matching version only performs at an adequate level when it is modified with some simple linguistically motivated heuristics. When we would compare the complexity of the three solutions in terms of code, the Porter based solution is by far the most simple solution. Also when we compare the amount of man hours spent on developing a full morphology for Dutch with the development time of the Dutch Porter stemmer, the choice would be Porter. The development time was about 6 person-weeks.

The main advantage of using a dictionary for stemming is that we can use it as a resource for a compound splitter. We found that compound splitting is very effective for Dutch ad hoc retrieval, especially when compound parts and the original compound form are added to the index. Apparently, this combination technique leverages the positive effects of having full compounds (good for precision) and compound parts (good for recall).

The fuzzy matching based solution on the other hand is probably useful without modification for a large number of western European languages. Fuzzy matching also offers matching with spelling and OCR errors for free, which is a considerable advantage in applications where part of the document database is scanned, or foreign proper names play an important role.

Precision and/or recall enhancement? In Kraaij & Pohlmann (1996b), the main research focus was to investigate whether stemming could enhance retrieval performance on a Dutch document collection. The experiment was especially focused at recall since stemming is usually seen as a recall enhancement technique. Results showed it was indeed possible to improve recall, but that precision was hurt at the same time. When the same

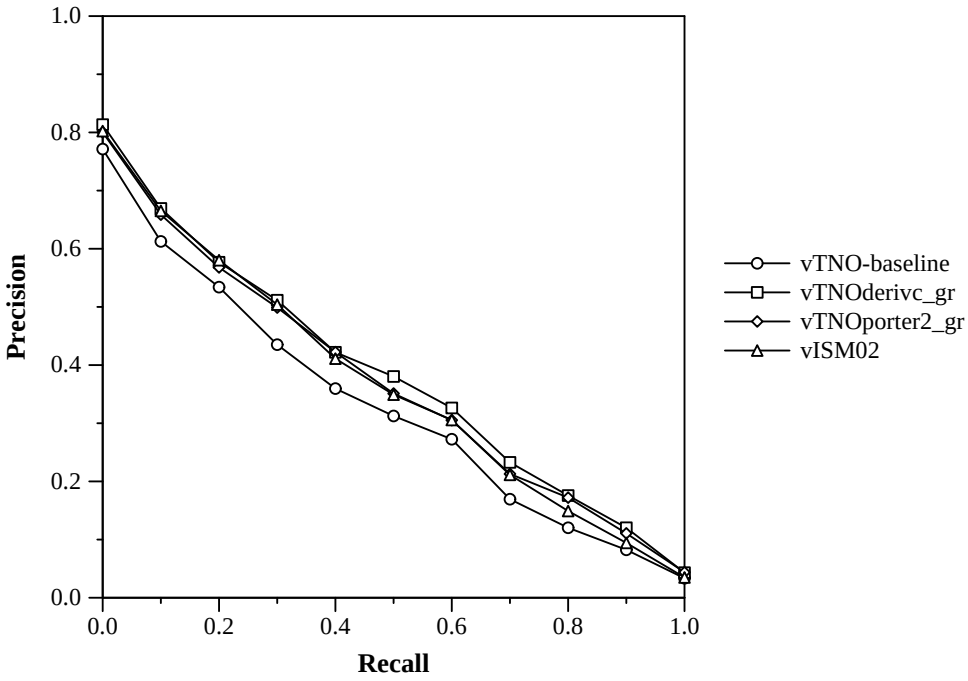

Figure 6.2. Comparison of the main stemming techniques

experiment was replicated using more advanced term-weighting algorithms, it was found that stemming can indeed improve recall and precision simultaneously. We will discuss these differences in performance in more detail in section 6.3.

Another method to gain understanding about the issue of whether stemming enhances recall or precision is to look at individual queries. Since mean average precision numbers are averaged across a query collection, they do hide a lot of detail (Hull, 1996). We therefore analyzed the differences between a plain run (no stemming) and a run with the "Dutch Porter" algorithm at the query level. Table 6.9 shows that the three measures

category	count
all +	28
all –	17
only ap5-15 –	7
only ap5-15 +	5
no change	4
only map +	2
only R-recall +	2
only R-recall –	1

Table 6.9. query level analysis of the impact of stemming for ap5-15, map and R-recall

ap5-15, mean average precision and R-recall are highly correlated. For less than a quarter of the cases there seems to be a recall-precision trade-off in the sense that either initial precision improves and mean average precision or R-recall decreases or the other way around. The main effect of stemming (for more than three quarter of the topics) seems to be that retrieval performance is either improved or hurt for all three measures simultaneously. This could be explained by the fact that the performance measures are highly correlated, which is a well known fact from meta-studies on TREC (cf. section 4.3.7). But we think it it also shows that stemming is generally affecting retrieval performance at all recall levels in a similar fashion. In most cases, stemming helps both recall and precision, in fewer cases stemming hurts both recall and precision. Since the performance improvements are usually larger than the performance decreases, the net effect of stemming is positive. In order to compute an upper bound of retrieval effectiveness of an IR system using stemming, an artificial run was constructed consisting of the maximum performance of the system with and without stemming for each query (after an idea in (Allan & Kumuran, 2003)). The mean average precision of this artificial run is 0.3675, which shows that the negative effects of stemming are relatively small.

6.3. Conflation architectures

In this section, we will discuss several approaches to combine morphological normalisation with retrieval models. As already mentioned in the previous section, we found significant differences in retrieval effectiveness of stemming using either the TRU engine (described in Kraaij & Pohlmann (1996b)) and the TNO engine (described in section 6.2). This suggests that it is very important that stemming is properly embedded into the retrieval model. We will first discuss various ways to combine stemming with retrieval models, either on-line or off-line (section 6.3.1). Several authors have remarked that stemming is a rather crude operation, since all morphological variants are treated equally. In section 6.3.2, we will propose and evaluate some more refined integration methods based on a language modeling framework. The methods use different knowledge sources and different metaphors to implement stemming within the retrieval model.

6.3.1. Off-line vs. Online stemming. The traditional way to use stemming for IR purposes is to reduce all the words in documents and queries to a base form and compute term-statistics on these base forms instead of the original words. For documents, this operation can be done at indexing time (off-line) or at retrieval time (on-line). Thus, stemming creates *equivalence classes*, all members of such a class are *conflated*. The challenge is to construct conflation methods that group words together that have the same meaning. Off-line stemming has the side effect that the number of postings for a document is reduced, making indexes more compact and retrieval more efficient.

6.3.1.1. *The advantages of on-line stemming.* It is possible though to implement stemming in an on-line fashion, as demonstrated by Harman (1991). The idea is to replace each query term by the members of its conflation class at retrieval time by merge the posting lists of the members of these classes (at retrieval time, for each query term). This has the result that term-frequencies of conflated terms are summed and document frequencies are based on the equivalence class rather than the original terms. Further details about the implementation are given in section 6.3.1.4.

On-line stemming has several important advantages. The main advantage of on-line stemming is its use for interactive systems. Since users want to understand why a system retrieved a document, it might be important for a user to have control over the conflation process. Stemmers are not perfect, and an on-line conflation approach makes it possible to remove unwanted expansions. E.g. our Dutch stemmer erroneously conflates *eerst (first)* and *Eersel (name of a village).* A second advantage is that stemming saves indexing time and index space: only a single inverted file is necessary instead of one inverted file per stemmer variant.

6.3.1.2. *Approximating on-line stemming by query expansion.* As already noted, Kraaij & Pohlmann (1996b) implemented the conflation step by query expansion in a first series of experiments with different stemmers. Query expansion seemed a practical solution, since it facilitated the comparison of many different stemmers in an environment with limited disk space and limited computing power. However, since the source code of the TRU search engine was not available, proper computation of the document frequency of conflated classes could not be realized. Since Harman had shown that query expansion plus re-weighting (see section 6.3.1.3) was just as effective, it was hypothesized that implementing morphological normalization by query expansion would not significantly hurt our results. Replication of the experiments (cf. section 6.3.1.4) showed that this assumption did not hold for more advanced term-weighting algorithms and more importantly that the main conclusions w.r.t. the effectiveness of stemming for Dutch as stated in Kraaij & Pohlmann (1996b) had to be revised.

It is important to look at global term statistics in order to understand the difference between full conflation and (naive) query expansion. Conflation is a form of normalisation; terms that are in the same conflation class are treated as if they were the same term. The conflation operation has important effects on global term statistics: the number of distinct terms in a document collection is reduced, whereas the within document frequencies of these conflation classes are higher than in the cases without conflation. The total number of index terms in a document does not change, but the token-type ratio changes, i.e. there are less unique terms. One could see this as a variance reduction operation, improving the estimates of relative frequencies of concepts in the documents (when we see conflation classes as some form of proto-concepts). So both the within-document term frequencies and the collection characteristics of term(-classes) are affected by conflation. That means that the term/class weights change as well, both in documents and queries. In a query expansion operation, global term statistics do not change, consequently morphological reduction by query expansion has different characteristics than reduction by conflation.

Let us look at an example query: "*Agressie op school* (agression at school)". If we only consider inflectional conflation, there is only one word with related morphological variants: *school.* After stopping and morphological expansion, the query would thus be transformed in: *agressie(248),school(2940), scholen(1607),schooltje(37),schooltjes(9),* the numbers between parentheses show the document frequency of these terms. This query expansion has two unintended side effects:

(1) Since the term *school* has been expanded, most statistical IR systems will put more emphasis on this concept, whereas in the original query, the emphasis

would be on *agressie* since this word has the highest idf of the original query terms.

(2) Documents containing the rare diminutive forms *schooltje* or *schooltjes* will rank very high, since these forms have a high idf.

One could remedy the first effect by normalising the query term weights of expanded terms such that the variant term weights sum up to one (just like in the QT model that we discussed in chapter 5), but it is not so easy to correct for the idf effect, without having access to the source code of a search engine.

6.3.1.3. *Reweighting expansion terms.* Previous research suggested that re-weighting expansion terms by assigning them a lower weight might help to improve the performance of stemming (Harman, 1991). The intuition is that in a lot of cases, the exact term match is a better indicator for relevance than the match with a morphological variant. We experimented with a variant approach: we re-weighted the original query terms by including them three times in the expanded query (these system versions have the suffix "ow"). This approach turned out to work well. We can illustrate that with the results in table 6.10 (taken from Kraaij & Pohlmann (1996b)). We compared a version doing Porter based stemming at indexing time (vp2pr) with a version based on (naive) query expansion (vp2) and a version where the expansion terms are re-weighted (vp2ow). We also compared naive query expansion and a re-weighted variant for dictionary based query expansion (vc1/vc1ow). We concluded several things from these results: (i) stemming by query expansion can be as efficient as normal stemming (during indexing) , when it is complemented with the re-weighting procedure (ii) the re-weighting procedure is consistently better than the naive expansion procedure and yields a very small improvement over the baseline system.

version	ap5_15	% change		map	% change		R-recall	% change	
vn	0.438			0.284			0.310		
vp2	0.348	−	21	0.229	−	19	0.261	−	16
vp2ow	0.444	+	1	0.299	+	5	0.319	+	3
vp2pr	0.442	+	1	0.294	+	4	0.312	+	1
vc1	0.307	−	30	0.220	−	23	0.244	−	22
vc1ow	0.446	+	2	0.292	+	3	0.315	+	2

Table 6.10. Comparison of naive versus structured query expansion (TRU engine)

6.3.1.4. *Implementation of on-line conflation.* The TNO engine enabled us to redo some of the experiments, because the engine supports a conflation operation operation at retrieval time, which computes the proper global statistics (idf) for the conflation class on-the-fly. The procedure works as follows:

(1) Replace each query term by a list of terms which form the conflation class, conflation classes can be constructed off-line using a stemmer and the vocabulary of index terms. The result query is structured like a list of conflation classes.

(2) The retrieval engine scores documents, by treating one conflation class at a time, for each conflation class do:

- Compute the idf of the conflation class, by counting the documents that contain at least one member of the conflation class *or* sum the collection frequencies of each member of a class for estimates based on the collection frequency.
- Score documents with respect to the conflation class by counting the total number of occurrences of all members of the conflation class, use this number as the within document frequency.

This procedure has the effect that documents will produce the same ranking as if the conflation procedure would have been carried out at indexing time[7].

The procedure to implement conflation by query expansion was inspired by Harman (1991). She called this on-line conflation process *grouping* since it effectively imposes a structure on the expanded query. In her experiments however, the idf values for the conflation classes (or concepts) were computed off-line and stored in a separate file. The INQUERY system supports a SYN operator. One can use this operator to construct conflation classes, which are evaluated in the same way as we described. The SYN operator has been recently applied successfully for query expansion (Kekäläinen & Järvelin, 2000) and cross language information retrieval (Pirkola & Järvelin, 2001). Since these publications, the "grouping" method has also become known as the "structured query" method. We refer to chapter 5 for a more detailed discussion of the use of the conflation operator for CLIR.

We replicated the experiments with several (approximations of) on-line stemming methods based on the TRU engine (cf. table 6.10) with a new series of experiments based on the TNO engine, results are presented in table 6.11. When we compare the results in

version	ap5_15	% change	map	% change	R-recall	% change
vTNO-plain	0.481		0.322		0.356	
vTNOp2-conflate	0.519	+ 8	0.362	+ 12	0.388	+ 9
vTNOp2-naive	0.394	− 18	0.236	− 27	0.272	− 24
vTNOc1-conflate	0.521	+ 8	0.375	+ 17	0.386	+ 9
vTNOc1-naive	0.309	− 36	0.231	− 28	0.267	− 25
vTNOc1-naive-ow	0.478	− 1	0.329	+ 2	0.365	+ 3

Table 6.11. Comparison of naive versus structured query expansion (TNO engine). "conflate" refers to on-line stemming, naive refers to naive query expansion, naive-ow to the variant where the original query terms receive a higher weight, p2 is the Dutch stemmer, c1 refers to dictionary based derivational stemming.

table 6.11 with the results in table 6.10, we see similarities: naive query expansion is not effective and down-weighting expansion terms helps to some degree. However, there is one big difference: the versions based on true conflation (idf statistics for the conflation

[7] The equivalence between on and off-line conflation does not hold for all term weighting models. E.g. conflation for term weighting procedures that rely on the average term frequency, number of unique terms (Lnu.ltu) or the sum of document frequencies (the original Hiemstra model) cannot be implemented efficiently in an on-line fashion, since these statistics would have to be recomputed for each document in the collection.

class are computed on the fly) perform much better than the baseline system and the re-weighted versions for both initial precision, and higher recall levels.

It is interesting that down-weighting the expansion terms helps for both the TRU and TNO engine based runs. We hypothesized that an IR model which includes a weighted translation model (cf. section 2.6.3.3 or chapter 5) might be a suitable framework to model and exploit this effect. We will investigate this in the next section.

Concluding, our early experiments with stemming based on on-line query expansion yielded sub-optimal results. However, it is possible to implement an on-line version of conflation which yields results that are equivalent to stemming at indexing time. We have used this on-line version of conflation for the experiments in section 6.2..

6.3.2. Modeling stemming in a LM framework. In section 6.3.1.2, we have seen that naive query expansion is not effective to implement on line stemming. We found that grouping term variants for proper collection frequencies is indeed an essential technique. On the other hand, stemming is known as a crude operation, which improves retrieval effectiveness of many queries but also hurts a significant proportion of the queries. We have confirmed this result by the short analysis in section 6.2.5. There are similar results for English. Both Harman (1991) and Hull (1996) found that stemming is not an effective technique averaged across queries. But for individual queries, performance can be significantly improved or hurt. The question whether stemming improves retrieval effectiveness for a particular language seems to be mostly determined by the ratio of queries that are improved or hurt. This result indicates that there is room for improvement. This improvement could for example be achieved by

- An adaptive system that determines whether stemming is necessary for a certain query or certain query terms.
- Weighted stemming: a stemmer that uses either external linguistic knowledge about semantic similarity between morphological variants or a corpus based similarity notion to restrict stemming to a restricted set of variants or to weight individual variants.
- An IR model which includes the amount of stemming as a (global) parameter.

Harman did several experiments to test some of these hypotheses. An experiment where stemming was restricted to short queries or to terms with a high idf value (i.e. rare terms) slightly degraded results. The term re-weighting experiment that we mentioned earlier, where expansion terms get a lower weight reached performance similar to on-line conflation. Thus it seems that assigning non-uniform weight at terms in a conflation class might help in finding a better trade-off between helping and hurting queries. Maybe postulating mere equivalence between all term variants which are conflated to the same base form by a stemmer is too simplistic. Riloff (1995) gives some evidence that in some cases even the distinction singular/plural is quite important.

More recently, Ponte suggested that the usual application of stemming for LM based IR might be too crude to show a gain in effectiveness for monolingual English IR. He hypothesized that a more refined embedding, using a mapping function different from the usual binary yes/no decision for conflation, would be an interesting area to explore (Ponte, 2001). These publications have inspired the research questions for this section.

resarch question. We will investigate whether we can refine the traditional approach of stemming for IR by evaluating some more complex models that exploit heuristic, linguistic and/or corpus based knowledge. The more complex models perform matching in wordform feature-space instead of word-stem feature-space. We will carry out this experiment using the generative probabilistic framework that we applied successfully for CLIR. In fact, some researchers state that monolingual retrieval is a special case of CLIR and CLIR models are very well suitable for monolingual IR, because provide an easy facility to integrate polysemy in the retrieval model (Berger & Lafferty, 1999; Xu et al., 2002a). We have tested five ideas to implement weighted stemming, which can be categorized in two classes. The first class (described in section 6.3.2.2) comprises three ways to implement re-weighting of expansion terms in a probabilistically sound manner. One of the aims of the experiment is to check whether a probabilistic version of Harman's idea yields better results. The first two variant models use heuristic parameter settings for the re-weighting procedure. The third model is in addition motivated by linguistic arguments. The idea is that inflectional variants have a higher probability to be similar than derivational variants and thus should be weighted different. In fact this test could be seen as a linguistically motivated instantiation of the first category. The second category (described in section 6.3.2.3) consists of two tests that use the document collection to estimate similarity between wordforms. All tests will perform matching in the event space of unstemmed terms. As an introduction we will give a simple formalization of a monolingual IR model, which includes stemming.

6.3.2.1. *Integrating stemming in a probabilistic IR model.* In fact integrating standard stemming into an IR model is almost trivial, since it is just a matter of making the equivalence classes explicit in the estimates for the individual language models. We start with our basic language model, which was presented in section 2.6.3.5:

$$CER(Q;C,D) = \sum_{i=1}^{n} P(\tau_i|Q) \log \frac{P(\tau_i|D)}{P(\tau_i|C)} \tag{81}$$

Here τ_i is defined as index term. An index term can be anything, e.g. a full wordform, a stem or a character n-gram. Now, for we define the following variables: ϕ_i represents a wordform and σ_i represents a stem class. If we want to work with a fully stemmed IR model, this can be achieved by estimating unigram stem models $P(\sigma_i|M)$ using the counts of ϕ_i. We already discussed the construction of equivalence classes in the context of LM based IR, in section 5.2.3 where we discussed the use of the synonym operator for CLIR. The synonym operator can be seen as a special case of a probabilistic (word by word) translation model, implemented by a convolution operation. Wordforms are translated into their stem class with a probability of one. All other translation probabilities are zero. More formal:

$$P(\sigma_i|D) = \sum_{j}^{\Phi} P(\phi_j, \sigma_i|D) \approx \sum_{j}^{\Phi} P(\sigma_i|\phi_j) P(\phi_j|D) \tag{82}$$

$$P(\sigma_i|\phi_j) = \begin{cases} 1 & \text{if} \quad \phi_j \in \sigma(s_i) \\ 0 & \text{if} \quad \phi_j \notin \sigma(s_i) \end{cases}$$

where Φ represents the total number of wordforms in the vocabulary of the collection. The approximation step, which states that stemming can be implemented by a "translation" step which is assumed to be independent from the language model concerned (e.g. a document or query) might be the very reason why stemming is not an unequivocal success for IR. The assumption is important though for efficiency reasons, since it simplifies the model enormously, because the (binary) translation weights $P(\sigma_i|\phi_j)$ can be implemented by a context insensitive morphological normalization method.

When we substitute the reformulated language model of formula (82) in formula (81) we arrive at:

$$CER(Q;C,D) = \sum_{i=1}^{n} \sum_{j}^{\Phi} P(\sigma_i|\phi_j)P(\phi_j|Q) \log \frac{\sum_j^{\Phi} P(\sigma_i|\phi_j)P(\phi_j|D)}{\sum_j^{\Phi} P(\sigma_i|\phi_j)P(\phi_j|C)} \tag{83}$$

which is quite similar[8] to model (78), the model for pivoted translation, where both query and document models are mapped into the pivot language, before matching takes place. In this case the translation takes place between two different index languages: an index of wordforms and an index of stems, which are both representations of texts written in the same natural language. Our challenge is thus to come up, either with a more refined translation model or to perform the matching process in the event space of wordforms and enrich the language models of query and or document by taking advantage of morphological knowledge. We have experimented with several variant systems that address this challenge.

6.3.2.2. *Weighted stemming based on heuristics.* In a follow-up experiment to improve stemming effectiveness, Harman experimented with down-weighting the expansion terms. The idea is that added terms are on average less good search terms than the original query terms. Unfortunately her test system did not allow down-weighting within an equivalence class, so she had to fall back on naive query expansion. Her experiments showed that down-weighting with a factor 2 gave a significant improvement over no grouping (plain naive query expansion) and equal the results for the grouping version.

We have replicated Harman's experiments on the UPLIFT collection, and we have investigated whether weight differentiation within an equivalence class or partitioning the equivalence class improves retrieval performance.

The first idea is to recast Harman's experiment in a LM framework. The least we can do is to normalize probabilities of term expansions. This means that if we expand a query term, we will make sure that the probability mass of the total expansion of that term does not change. We tested two variants, a variant with unweighted expansion (vc1-naive-eq) and a variant where the expansion terms received half the weight of the original term (vc1-naive_ow-q). Both variants effectively transform the language model of the query into a richer model, using linguistic information. The second idea is to apply the same procedure at the side of the document language model. The weighted

[8]We omitted smoothing for presentation reasons. Also unlike model (78), query and document representation languages are equivalent here.

expansion of a query language model can be formalized as follows:

$$P(\phi_i|Q) = \sum_j^{\Phi} P(\phi_i|\phi_j) P_{ml}(\phi_j|Q) \tag{84}$$

$$P(\phi_i|\phi_j) = \begin{cases} \nu/(\nu + |\sigma(\phi_j)|) & \text{if} \quad \phi_i = \phi_j \\ 1/(\nu + |\sigma(\phi_j)|) & \text{if} \quad \phi_j \in \sigma(\phi_j) \\ 0 \quad \text{if} \quad \phi_j \notin \sigma(\phi_j) \end{cases}$$

where ν is the up-weighting factor of original query terms. The weighted conflation of a document language model can be formalized as follows:

$$P(\phi_i|D) = \sum_j^{\Phi} P(\phi_i|\phi_j) P_{ml}(\phi_j|D) \tag{85}$$

$$P(\phi_i|\phi_j) = \begin{cases} \nu/(\nu + |\sigma(\phi_i)|) & \text{if} \quad \phi_i = \phi_j \\ 1/(\nu + |\sigma(\phi_i)|) & \text{if} \quad \phi_j \in \sigma(\phi_i) \\ 0 \quad \text{if} \quad \phi_j \notin \sigma(\phi_i) \end{cases}$$

A quite similar experiment was independently carried out by Allan & Kumuran (2003). They formalized the idea by a mixture model. Despite the fact the probabilities in their model do not sum to one, their model implements the same idea.

The third idea for a more refined approach to stemming was based on the intuition that derivational variants are often more remotely related to the original query terms. These terms also have quite different collection statistics. We hypothesized that it might be better to restrict steming to inflectional variants, but to expand with derivational variant stem classes. We found that a down-weighting factor of 10 was most effective. Formula (86) formalizes the procedure:

$$P(\phi_i|Q) = \sum_j^{\Phi} P(\phi_i|\phi_j) P_{ml}(\phi_j|Q) \tag{86}$$

$$P(\phi_i|\phi_j) = \begin{cases} 1/\mu & \text{if} \quad \phi_j \in \sigma_{\text{deriv}}(\phi_j) \\ 1 & \text{if} \quad \phi_j \in \sigma_{\text{infl}}(\phi_i) \\ 0 & \text{if} \quad \phi_j \notin \sigma\text{inf}(\phi_i) \wedge \phi_j \notin \sigma\text{deriv}(\phi_i) \end{cases}$$

Here, $\sigma_{\text{deriv}}(\phi_j)$ is the set of derivational variants of ϕ_j and their inflectional variants. $\sigma_{\text{infl}}(\phi_j)$ represents the equivalence class of inflectional variants of ϕ_j. μ is the down-weight factor. Examples of the expansions of a single query term by the different expansion models can be found in table 6.12. Results of the three weighted stemming models and the four baselines (no expansion, stemming, naive expansion, normalized naive expansion) are presented in table 6.13. Results are disappointing. The latter two models (85) (86) (which both include a conflation component) perform minimally better than the standard stemming baseline based on conflation. We did not perform significance tests, since the difference in mean average precision is not of practical interest. The first model (84), based on re-weighted query expansion is indeed able to improve upon uniform normalized query expansion, which in turn performs much better than 'naive' query expansion. However, it seems that true conflation is more effective for this

run id.	explanation	transformed query
vLM-plain	none	kunstmatig:1
vc1	grouping	(kunstmatig:1; kunstmatige:1; kunstenaar:1; kunst:1)
vc1-naive	naive expansion	kunstmatig:1; kunstmatige:1; kunstenaar:1; kunst:1
vc1-naive-eq	normalized expansion	kunstmatig:0.25; kunstmatige:0.25; kunstenaar:0.25; kunst:0.25
vc1-naive-eq	normalized reweighted expansion	kunstmatig:0.4; kunstmatige:0.2; kunstenaar:0.2; kunst:0.2
vc1-w-0.5	reweighted grouping	(kunstmatig:0.4; kunstmatige:0.2; kunstenaar:0.2; kunst:0.2)
vc1d10	downweight derivation only	(kunstmatig:1; kunstmatige:1) ; (kunstenaar:0.1) ; (kunst:0.1)

Table 6.12. Example of transformations of the query 'kunstmatig'. Parentheses mean that the terms belong to a single equivalence class and are evaluated by (weighted) conflation.

(Dutch) test collection than heuristically re-weighted query expansion with morphological variants. It is disappointing that the big gain due to re-weighting for query expansion is only very small, when implemented as weighted conflation.

version	ap5_15	% change	map	% change	R-recall	% change
vLM-plain-cf	0.453		0.301		0.335	
vc1	0.495	+ 9	0.346	+ 15	0.358	+ 7
vc1-naive	0.259	− 43	0.169	− 44	0.210	− 37
vc1-naive-eq	0.421	− 7	0.285	− 5	0.314	− 6
vc1-naive-ow2-eq	0.466	+ 3	0.321	+ 7	0.334	− 0
vc1-w-0.5	0.499	+ 10	0.348	+ 16	0.370	+ 10
vc1d10	0.499	+ 10	0.351	+ 16	0.367	+ 10

Table 6.13. Results of weighted stemming based on heuristics

6.3.2.3. *Using the corpus to refine stemming.* We experimented with two other models, where we used the document collection as a resource for determining whether two morphological variants are related. One model involves a simple translation step, the idea is that we could model the morphological expansion process as a translation step just like the CLIR models we presented in the previous chapter. The second idea is that we try to use the document collection for query expansion, but guided by linguistic constraints.

In the previous chapter, we presented the QT and DT for CLIR, where either the query language model was projected into the document language using convolution with a translation matrix (QT) or vice versa (DT). For CLIR, we constructed translation models

using word aligned parallel corpora. We already saw that we could use a parallel corpus to improve monolingual retrieval (cf. section 5.5.6), but these experiments were defined on stemmed collections. We tried something simple for this experiment, namely to use co-occurrence in the document collection itself as a means to estimate $P(\phi_i|\phi_j)$. The intuition is that we already know which terms are candidate for expansion (the morphological variants) but that we want to down-weight expansion terms that are hardly related to the query term (e.g. *kunstmatig* , *kunstenaar* (=artificial, artist)). This can be achieved by a translation step. The "translation" probabilities are estimated as the conditional probability of ϕ_i co-occurring with ϕ_j in a document:

$$P(\phi_i|\phi_j) = \frac{c_d(\phi_i, \phi_j)}{c_d(\phi_j)} \tag{87}$$

where $c_d(\phi_j)$ is the number of documents that contain the wordform ϕ_j and $c_d(\phi_i, \phi_j)$ is the number of documents that contain both ϕ_i and ϕ_j. The estimates were smoothed using a fixed back-off conditional probability of 0.01. Of course, if $\phi_i = \phi_j$, the translation probability is 1. We know that first order co-occurrence is generally considered too weak to use it as a basis for locating related terms, but since the candidate terms are already morphologically related, first order co-occurrence might be a sufficient source of evidence.

The second corpus based approach is to estimate a weighted model of the expanded query using the *relevance model* technique developed by Lavrenko & Croft (2001). The basic retrieval model of Lavrenko and Croft is rather similar to ours, in the sense that it is based on measuring the cross entropy between two language models: a language model for each document and a relevance model. Lavrenko and Croft propose several ways to estimate relevance models either with or without explicit information about relevant documents (Lavrenko & Croft, 2003). The basic idea for estimating a relevance model without relevance information is to use just the query terms as a starting point and to estimate the joint probability by summation over the universe of language models. In practice usually a subset of documents is taken to represent the universe. We will present the Lavrenko and Croft model in a notation, which is in line with the other models of this section, where a wordform is represented as ϕ. q_i are the original unstemmed query terms.

$$P(\phi_i|Q) = P(\phi_i|q_1, q_2, ..., q_n) = \frac{P(\phi_i, q_1, q_2, ..., q_n)}{P(q_1, q_2, ..., q_n)} \tag{88}$$

Now we can estimate the joint probability by summing over the documents and assuming conditional independence:

$$P(\phi_i, q_1, q_2, ..., q_n) = P(\phi_i) \sum_{D_j \in C} P(D_j) P(\phi|D_j) \prod_{k=1}^{n} P(q_k|D_j) \tag{89}$$

The usual procedure of relevance modeling is to estimate $P(\phi_i|R)$ for each term in the indexing vocabulary and subsequently measure cross entropy with all document language models. The disadvantage of the method is that this can be quite inefficient. We restricted computation to just the query terms themselves and their morphological variants and normalized in order to yield a proper $P(\phi_i|R)$ and subsequently proceeded as usual. Best results were obtained with rather strongly smoothed models and taking

the top 50 documents of an initial query for the summation in formula (89). Table 6.14

version	description	ap5_15	map	R-recall
vLM-plain-cf	no stemming	0.453	0.301	0.335
vc1	uniform stemming	0.495 + 9	0.346 + 15	0.358 + 7
vc1-qm	QT model	0.477 + 5	0.317 + 5	0.352 + 5
vc1-dm	DT model	0.478 + 6	0.329 + 9	0.358 + 7
rm-900-50	relevance model	0.499 + 10	0.343 + 14	0.367 + 10

Table 6.14. Results of weighted stemming using corpus based estimation

presents the results of the experiments with weighted stemming using the translation approach and the relevance model approach. Results are good, but not convincing, since they do not provide a gain w.r.t. standard conflation. Still, it is shown that matching in the event space of wordforms can reach the same level of retrieval effectiveness as matching in the event space of stems.

6.3.3. Discussion. In this section we have reviewed several alternative ways to integrate morphological normalization in a LM-based retrieval framework. Previous results had suggested that weighted stemming could improve upon the standard method, where stemming is merely used as a preprocessing step and retrieval is fully carried out in the event space of stems. Each morphological variant is weighted equal, which is not always optimal. We have proposed several models that assign lower weights to the various expansion terms either based on heuristics or corpus-based probability estimates, also in combination with linguistic knowledge. Most of these alternative models match in the space of wordforms. Some of the models reach the same performance level as stemming, but there is no significant improvement. This is disappointing, since many of the models contain parameters that have been tuned, so performance on a separate test-collection could be lower. We think that the fact that simple unweighted stemming is robust and hard to improve upon is due to the fact that matching takes place in a reduced event space (stems versus wordforms). This means that estimates for the parameters in the language models are more robust, since small sample variance is reduced (Ponte, 2001). The reduced event space indeed also introduces bias error, but the net benefit is clearly positive. Conclusions about the ineffectiveness cannot be definite. Experiments with other collections and especially short queries are necessary. It might be that the queries in the UPLIFT query collection are too long to show differentiated results between methods.

6.4. Overall conclusions

In this chapter, we proposed and evaluated methods for morphological normalization for Dutch IR. The methods all have a linguistic motivation, but the implementation level of morphological knowledge varies from minimalist to a complete dictionary. All methods are able to improve the baseline retrieval effectiveness significantly. Best results were achieved by the dictionary based method, which increased mean average precision with

15%. The Dutch version of the Porter algorithm achieves a comparable performance and is therefore a practical solution, since it is a small, fast and freely available module. Stemming by fuzzy matching, is also quite effective. This method has the advantage that it is language independent. Compound splitting is also an effective procedure. Adding compound parts to documents and query resulted in a significant extra gain in terms of mean average precision.

We did a very short analysis of the effectiveness of stemming at the individual query level and found that - contrary to what is usually assumed - stemming usually operates at a broad spectrum of all recall levels and thus is not only active at higher recall levels. The same analysis revealed that the room for improvement of weighted stemming is rather small. An artificial run based on the maximum mean average precision per run of the stemmed and unstemmed runs yielded only about 1% absolute performance improvement.

In the second part of the chapter, we reviewed several (on-line) stemming models. Several authors had suggested that a more refined approach to stemming could be more effective than the standard approach where variants are replaced by their stem. We constructed several LM based IR models where stemming was integrated in several different ways in the estimation procedures. The methods all attempted to realize a more refined way to include morphological variants in the document ranking process. Methods exploited different knowledge sources and were based on different metaphors, inspired by our own work on CLIR and the work of Lavrenko. None of these methods succeeded in improving upon the basis stemming method. We think that this is due to the variance/bias trade-off. The more refined conflation models operate in the event space of wordforms instead of stems, which is more sparse. Consequently there is far more sample variance. So far, we have not been able to reduce sample variance in the wordform event space effectively, in order to benefit from the reduced bias error of the wordform space. Future experiments could help to sharpen understanding of this issue. The sparseness of wordform based document models might be overcome by document-specific smoothing (cf. section 3.1.6. Experiments with different test collections and different languages could help to gain a better judgement whether the observed effects can be generalized.

CHAPTER 7

Score normalization for topic tracking

Generative unigram language models have proven to be a simple though effective model for information retrieval tasks. Such IR models assign a matching score (RSV=retrieval status value) to a document, which reflects its degree of relevance with respect to a certain topic. The scores can usually not be interpreted on an absolute scale, since several approximations and simplifications preclude interpretation as a pure probability value. This is not a problem for ad hoc retrieval tasks where scores are only used to produce a rank order and not to evaluate performance in an absolute sense. The ranking process for a certain topic is completely independent of other topics.

In contrast to ad hoc retrieval, there are IR-tasks that do require matching scores that are comparable across topics. An example of such a task is the topic tracking task as defined for the TDT (Topic Detection and Tracking) benchmark evaluation. The task definition requires that matching scores are comparable on an absolute scale, since documents are filtered using a global score threshold. A second application where score comparability plays a role is cross-lingual search in a multilingual document collection. In order to yield a merged ranked list of retrieved documents in different languages, scores have to be normalized across languages. In this chapter, we will investigate several ways to normalize scores in the context of a topic tracking application.

7.1. Introduction to Tracking

Topic tracking is one of the tasks of the annual Topic Detection and Tracking (TDT) evaluation workshop, which was first organized in 1996. Main purpose of the TDT project is to advance the state-of-the-art in determining the topical structure of multilingual news streams from various sources, including newswire, radio and television broadcasts, and Internet sites. See (Wayne, 2000) for a detailed overview of the TDT project. The tracking task models the information need of a user who hears about a certain event on the radio or television and wants to be notified about all follow-up stories in a number of pre-specified information sources in different languages. TDT is challenging because it combines several problems: automatic speech recognition and segmentation of continuous media like radio and television, cross-lingual access to data and a topic tracking task without supervised relevance feedback. A topic tracking system is initialized with one or a few stories describing a certain news event, and must track this topic in a stream of new incoming stories. A tracker has to make binary decisions: a story is either on-topic or off-topic. In practice such a decision is based on thresholding a score which is designed to be some monotonic function of the probability that the story is on-topic.

The goal of this study is to investigate whether generative probabilistic models that have been successfully applied to ad hoc IR tasks in TREC (cf. e.g., Hiemstra, 1998; Miller et al., 1999a; Berger & Lafferty, 2000; Ng, 2000a; Hiemstra & Kraaij, 1999) can be applied to the tracking task and how these approaches should be adapted in order to generate normalized scores. In the following sections, we will review several ways to use generative models for tracking and methods to obtain comparable scores across topics in order to identify a single model which is effective for both the ad hoc and tracking task.

The discussion of score normalization in a tracking context is organized into three main sections. Section 7.2 compares several lay-outs for the use of language models for ad hoc IR and topic tracking, in particular, we will look at model-internal and external normalization methods. In section 7.3 we describe experiments with a selection of models that have been carried out on the TDT development data and on the TREC-8 ad hoc data. The tracking study will be concluded with a discussion.

7.2. Language models for IR tasks

The basic problem underlying most information retrieval tasks is that of ranking documents based on relevance with respect to a certain information need. The object of interest can be an ad hoc topic, a long-standing topic of interest or - in a more dynamic fashion - an event of interest. An implicit requirement is that document ranking functions need to be able to cope with documents of different lengths. In some of the TREC collections for example, document sizes can differ several orders of magnitude. If a score would be correlated with document length, this would cause highly inflated scores for long documents. For some IR tasks like topic tracking or distributed IR, simple ordering is not enough. In the TDT tracking task that we will study in this chapter, matching scores have to interpretable on an absolute scale, since the score is used to classify a document as relevant or not relevant using a global threshold value. As a consequence scores must be comparable across stories (documents) and topics (queries). For certain applications (e.g. document clustering) it is even desirable that matching scores fulfil another constraint, namely symmetry (Spitters & Kraaij, 2002), since clustering algorithms presuppose a symmetric similarity function.

7.2.1. Score properties of probabilistic models. Language models have been applied with success for topic tracking by BBN (Leek et al., 2002). Both their 'TS' (topic spotting) and 'IR' model[1] for topic tracking are effective, but it seems that the score distribution properties of the 'TS' and 'IR' model and also the relationship between these models is not completely understood. We therefore review the relationship of classical probabilistic models and generative probabilistic models for IR with regard to the aspect of score normalization. For reasons of legibility, we will present the models from the point of view of an ad hoc IR problem, i.e. we talk about documents and queries. In most cases - unless stated otherwise - the models also apply to the tracking task, after replacing queries (Q: derived from a short description of an ad hoc information need) by topics (T: one or more training stories)[2].

[1] These are the actual model names that BBN uses in their publications

[2] It is customary to use Q and D in IR models and T and S in articles about TDT applications.

The following definitions stem from Sparck Jones et al. (2000):

- Q is the event that the user has a certain information need and describes it with description Q.
- D is the event of evaluating a document with description D
- L is the event that D is liked (or relevant).
- $\bar{L}$ is the event that D is not liked (or relevant).

Now, for a certain query, we want to rank documents on the probability that they are "liked". This can be done by estimating $P(L|D,Q)$: the probability that a document is liked given its description plus the description of the query. In order to simplify further computation[3], documents are ordered by log-odds of being liked, which is an order preserving operation.

7.2.1.1. *Document likelihood.* The classical next step (cf. section 2.6.1.1) is to apply Bayes' rule in order to express the matching score based on log-odds in terms of $P(D|L,Q)$ i.e. the probability that a document is described by D when we know it is relevant for a certain query Q. This model describes the situation consisting of one query and several documents.

$$\log \frac{P(L|D_i,Q)}{P(\bar{L}|D_i,Q)} = \log \frac{P(D_i|L,Q)}{P(D_i|\bar{L},Q)} + \log \frac{P(L|Q)}{P(\bar{L}|Q)} \tag{90}$$

Since no information about the prior probability of relevance given a certain query is available, a uniform prior is assumed. Therefore the second term in (90) can be dropped. Ranking is then solely based on the log-likelihood ratio $P(D_i|L,Q)/P(D_i|\bar{L},Q)$. One could interpret the numerator of the log-likelihood ratio as follows: "How likely is the description of document D_i if we assume the document is relevant to Q?". This likelihood is normalized by the likelihood of the document description given a model based on descriptions of non-relevant documents[4]. Because the model is about one query and several documents, scores are inherently comparable across documents.

Now $P(D_i|L,Q)$ (and $P(D_i|\bar{L},Q)$) can be estimated in various ways. In the Binary Independence Model (Robertson & Sparck Jones, 1976), also known as the Binary Independence Retrieval (BIR) model (Fuhr, 1992), D is described as a vector of binary features x_k, one for each word in the vocabulary. Further development of the log-odds assuming term independence leads to the classical Robertson-Sparck-Jones formula for term weighting. Estimation of $P(D_i|L,Q)$ is usually based on the assumption that there is prior knowledge about some relevant documents. The matching score based on the log-likelihood is basically a sum of term weights over all terms in the vocabulary, but usually it is assumed that $P(x_k|L,Q) = P(x_k|\bar{L},Q)$ for all terms that do not occur in the query. This means that scores of the 'typical' BIR model are comparable for documents, but not comparable for queries, since scores depend on the query length and not on document length. The BIR model can thus remain unchanged for the ad hoc IR task, but scores have to be normalized for topic length, if we would want to use the BIR model for tracking.

One can also estimate $P(D_i|L,Q)/P(D_i|\bar{L},Q)$ in a generative framework with D_i defined as a sequence of terms (cf. chapter 2 section 2.6.3). In such a generative framework

[3]The logarithm converts products to summations, working with the odds results in a simple likelihood ratio after applying Bayes' rule.

[4]This normalization is in fact the probabilistic justification of *idf* weighting

we can think of $P(D_i|L,Q)$ as the probability that D_i is generated as a sequence of terms from a unigram language model $P(w|R)$ which is constrained by Q and L i.e. which describes the probability of observing words in documents relevant to Q. As usual, term independence is assumed. This particular model is also referred to as 'document-likelihood' (Croft et al., 2001a). In a similar way we can think of $P(D_i|\bar{L},Q)$ as the probability that D_i is generated from a model estimated on non-relevant documents, which we can approximate by a model of the collection: $P(w|\bar{R}) \approx P(w|C)$. Since the vast majority of documents is not relevant, this seems to be a reasonable assumption. Substituting the generative estimates in the log-likelihood ratio results in:

$$\log \frac{P(D_i|L,Q)}{P(D_i|\bar{L},Q)} \approx \sum_{w \in D_i} d_w \log \frac{P(w|R)}{P(w|C)} \tag{91}$$

where d_w is the term frequency of the word w in the document. Just like the BIR Model, it is difficult to estimate $P(w|R)$ for ad hoc queries in the absence of relevance information. Applying maximum likelihood estimation on a short query would yield a very sparse language model. However, recently a new estimation technique has been developed to estimate $P(w|R)$ in a formal and effective way (Lavrenko & Croft, 2001). The so-called "relevance model" technique is based on estimating the joint distribution $P(w,Q)$ by making use of term co-occurrence in the document collection. For tracking, estimation is easier, since there is at least one example story. Stories are usually considerably longer than a typical ad hoc query.

Regarding score comparability, the situation is reversed in comparison with the BIR model. Scores are independent of query length (a relevance model is a probability distribution function over the complete vocabulary), but dependent on the length of the generated text, as can be seen in formula (91). We can illustrate this by comparing the scores of a document A and a document B, which consists of two copies of document A. Intuitively, both documents are equally relevant, but this is not reflected in the score. A simple correction is to normalize by document (story) length, making the score usable for both ad hoc and tracking tasks. It is interesting to note that a ratio of length normalized generative probabilities can also be interpreted as a difference between cross-entropies:

$$\begin{aligned} \sum_w \frac{d_w}{\sum_w d_w} \log \frac{P(w|R)}{P(w|C)} &= \sum_w \log P(w|D_i) \log P(w|R) \\ &\quad - \sum_w \log P(w|D_i) \log P(w|C) \\ &= H(D,R) - H(D,C) \\ &= CER(D;R,C) \end{aligned} \tag{92}$$

Here $P(w|D_i)$ is a unigram model of document D_i, which is constructed on the basis of maximum likelihood estimation. The basic ranking component in (92) is the (negated) cross-entropy $H(D_i,R)$, which is normalized by the cross-entropy $H(D_i,C)$. For relevant documents, the former cross-entropy will be smaller than the latter, so we can also refer to (92) as the cross-entropy reduction formula, since we are looking for documents that achieve a large entropy reduction w.r.t. the background language model. In previous publications, we referred to formula (92) as the (length) normalized log likelihood ratio formula (with shorthand $NLLR(D;Q,C)$). In this thesis we will use the shorthand

$CER(D;R,C)$ for a cross-entropy reduction ranking formula (cf. section 2.6.3.5 and the next section for a reversed direction of this formula) .

7.2.1.2. *Query Likelihood.* Coming back to the original log-odds model, Bayes' rule can also be applied to derive a model where the log-odds of being relevant are described in terms of $P(Q_j|L,D)$, i.e. the probability that a query is described by Q_j when we know that a document described by D is relevant (Fuhr, 1992; Lafferty & Zhai, 2001b).

$$\log \frac{P(L|D,Q_j)}{P(\bar{L}|D,Q_j)} = \log \frac{P(Q_j|L,D)}{P(Q_j|\bar{L},D)} + \log \frac{P(L|D)}{P(\bar{L}|D)} \tag{93}$$

Strictly spoken, this model describes the situation where there is one document plus a number of queries submitted to the system. Still, the model can be used for document ranking provided that the document models are constructed in a similar manner and do not depend on document length. In this case, the likelihood ratio computes how typical the query description Q_j is for document D in comparison to other query descriptions. Key element for the comparability of scores of different queries is the normalizing denominator $P(Q_j|\bar{L},D)$. Again, there are multiple ways to estimate $P(Q_j|L,D)$. A query representation by a binary feature vector leads to the Binary Independence Indexing (BII) model (Fuhr, 1992), which is closely related to the first formulated probabilistic IR model of Maron and Kuhns (Maron & Kuhns, 1960). Because of estimation problems, these models have - to our knowledge - not been used for practical IR-tasks like ad hoc queries or tracking. With regard to score comparability, these models should be normalized for query length in order to be used for tracking, the models can be applied unchanged for ad hoc tasks.

The query-likelihoods in (93) can also be estimated in a generative framework. One could think of $P(Q_j|L,D)$ as the probability that Q is generated as a sequence of terms from a model which is constrained by D and L. This means that a document with description D, which is known to be relevant, can be used as the basis for a generative language model which generates the query. Analogously to the document-likelihood model (91), we assume term independence and approximate $P(Q_j|\bar{L},D)$ by the marginal $P(Q)$:

$$\log \frac{P(L|D,Q_j)}{P(\bar{L}|D,Q_j)} = q_w \sum_{w \in Q_j} \log \frac{P(w|D)}{P(w|C)} + \frac{P(L|D)}{P(\bar{L}|D)} \tag{94}$$

where q_w is the number of times word w appears in query Q. The prior probability of relevance should not be dropped this time, since it can be used to incorporate valuable prior knowledge, e.g. that a Web page with many in-links has a high prior probability of relevance (Kraaij et al., 2002). This model is directly usable for the ad hoc task, since scores are comparable across documents of different lengths, due to the maximum likelihood procedure, which is used to estimate the language model $P(w|D)$. Usually, the denominator term $P(Q_j|\bar{L},D)$ is dropped from the ranking formula, since it does not depend on a document property.

$$\log P(L|D,Q_j) = P(L|D) \sum_{w \in Q_j} q_w \log P(w|D) \tag{95}$$

Equation (95) is the basic language modeling approach for ad hoc IR as formulated in (Hiemstra, 1998) and (Miller et al., 1999b).

However, if we want to use model (94) for tracking, scores should be comparable across queries. Therefore the denominator, which depends on the query, can not be dropped. In addition, scores have to be normalized for topic (=query) length[5], which leads again to a ranking formula based on the difference between two cross-entropies (for simplicity, we assume a uniform prior and drop the prior odds of relevance term in (94)):

$$\frac{q_w}{\sum_w q_w} \sum_w \log \frac{P(w|D)}{P(w|C)} = \sum_w \log P(w|Q_j) \log P(w|D) - \sum_w \log P(w|Q_j) \log P(w|C) = CER(Q;C,D) \tag{96}$$

Here, the basic ranking component is the reduction in cross-entropy that is achieved by comparing cross-entropy $H(Q,D)$ with the cross-entropy $H(Q,C)$.

Concluding, the probabilistic formulation of the prototypical IR-task, namely $P(L|D,Q)$, can be developed in two different ways: either by starting from documents, or by starting from queries. After applying Bayes' rule and transforming $P(L|D,Q)$ to log-odds, both variants can be rewritten to a sum of a likelihood ratio and the odds of the prior probability. The denominator in the likelihood ratio is a key element to ensure comparable scores of the events which are compared (document descriptions in the case of the document likelihood variant and query descriptions in the case of the query likelihood variant). Apart from the fact that the likelihoods of documents and queries have to be normalized in order to model $P(L|D,Q)$ (Bayes' rule), some corrections have to be applied to account for differences in length, since the basic model is based on descriptions of similar length.

The length normalization aspects of the various models are summarized in table 7.1. This table lists 'yes' if the particular model inherently accounts for length differences and 'no' if an external length normalization is required.

Model	query length normalization	document length normalization	reference
BIR	no	yes	Robertson & Sparck Jones (1976)
document likelihood ratio	yes	no	Lavrenko & Croft (2001)
BII	yes	no	Maron & Kuhns (1960); Fuhr (1992)
query likelihood (ratio)	no	yes	Hiemstra (1998); Miller et al. (1999b)

Table 7.1. Score properties of probabilistic IR models

[5]Matching scores are already length normalized in the language model of Ponte and Croft, since queries are represented as a binary vector defined on the complete collection vocabulary (Ponte & Croft, 1998).

7.2.2. A single model for ad hoc IR and tracking? As said, from an abstract point of view on matching, there are no major differences between the tracking and ad hoc task. There is some text, which describes the domain of interest of the user, and subsequently a list of documents has to be ranked according to relevance to that description. It seems valid to hypothesize that a good tracking system will work fine for ad hoc tasks as well, since the additional constraint concerning score normalization across topics does not affect the rank order of the documents. However, when we compare the tasks in more detail, there are certainly many differences between the ad hoc and the tracking task.

First of all, there is the dissimilarity of objects to be matched for the ad hoc task: a query is usually very short in comparison with a document. Moreover, not all words in the query are about the domain of interest, some serve to formulate the query. There are phrases like "Relevant documents discuss X" in ad hoc topics but not in TDT topics. In fact, the tracking task does not provide any query at all, just one or more example stories. In that respect, "matching" is much more symmetric for tracking. The asymmetry of the ad hoc task is probably the reason why the query likelihood approach is so successful: a document contains a much larger foothold of data to estimate a language model than a query (Lafferty & Zhai, 2001b). This preference is probably not so clear-cut for tracking. Indeed, BBN has experimented with both directions and found that they complement each other (Jin et al., 1999).

The ad hoc and tracking task also differ in their use of feedback techniques. Although the tracking task lacks supervised relevance feedback, unsupervised feedback (topic model adaptation) is allowed. In a way, this procedure is related to pseudo-feedback techniques in IR. However, the tracking task lacks the notion of the "top-N" documents, i.e. unsupervised feedback has to be based on absolute instead of relative scores, which is certainly more complicated.

In our experiments, we do not want to rule out specific models a-priori on the basis of the differences between ad hoc and tracking, but instead will investigate whether probabilistic language models which are successful for the ad hoc task can be adapted for tracking. We will study the necessity and relative effectiveness of normalization procedures. Therefore we will test both directions of the generative model for the tracking task.

7.2.3. Ranking with a risk metric: KL divergence. Recently, Lafferty and Zhai proposed a document ranking method based on a risk minimization framework (Lafferty & Zhai, 2001b). As a possible instantiation of this framework, they suggest to use the relative entropy of Kullback-Leibler (KL) divergence between a distribution representing the query and a distribution the document $KL(Q||D)$ as a loss function. The KL divergence is a measure for the difference between two probability distributions over the same event space.

$$KL(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{97}$$

KL divergence has an intuitive interpretation, since the KL divergence is either zero when the probability distributions are identical or has a positive value, quantifying the difference between the distributions by the number of bits which are wasted by encoding events from the distribution P with a "code" based on distribution Q. However, KL also

has some less attractive characteristics: it is not symmetric and does not satisfy the triangle inequality and thus is not a metric (Manning & Schütze, 1999).

The relationship between KL divergence and language models for IR was initially discussed by Ng (Ng, 2000a). The relationship of (96) with $KL(Q||D)$ is as follows:

$$KL(Q||D) = \sum_w P(w|Q) \log \frac{P(w|Q)}{P(w|D)} + \sum_w P(w|Q) \log \frac{P(w|C)}{P(w|C)} \tag{98}$$

which can be reformulated as:

$$CER(Q;D,C) = KL(Q||C) - KL(Q||D) \tag{99}$$

It is tempting to interpret this equation as a subtraction of two values on a particular measuring scale. However, this is invalid. Informally, we might interpret the cross-entropy reduction formula by taking a closer look at the two components: a score based on the cross-entropy reduction (CER) is high when $KL(Q||C)$ is high and $KL(Q||D)$ is low. This means that a story has a higher score when it contains specific terminology, i.e. is dissimilar from the background collection model and when its distribution is close to the topic distribution. For ad hoc search, $KL(Q||D)$ is essentially equivalent to the length normalized query likelihood (95) since the query entropy $H(Q) = \sum_w P(w|Q) \log P(w|Q)$ is a constant which does not influence document ranking. Several authors have presented KL divergence as a valid and effective ranking model for ad hoc IR tasks (Ogilvie & Callan, 2001; Lavrenko et al., 2002b). They consider query likelihoods as a derived form of the more general KL divergence. Since we are looking for a general model which is useful for both the ad hoc and the tracking task, we will evaluate the KL-divergence measure for tracking in addition to the models presented in section 7.2.

7.2.4. Parameter estimation. In the previous sections, we have only marginally talked about how unigram language models can be estimated. A straightforward method is to use maximum likelihood estimates, but just like language models for speech recognition, these estimates have to be smoothed. One obvious reason is to avoid to assign zero probabilities for terms that do not occur in a document because the term probabilities are estimated using maximum likelihood estimation. If a single query term does not occur in a document, this would amount to a zero probability of generating the query. There are two ways to cope with this. One could either model the query formulation process with a mixture model based on a document model and a background model or assume that all document models can in principle generate all terms in the vocabulary, but that irrelevant terms are generated with a very small probability.

A simple yet effective smoothing procedure, which has been successfully applied for ad hoc tasks is linear interpolation (Miller et al., 1999b; Hiemstra, 1998). Recently other smoothing techniques, namely Dirichlet and absolute discounting, have been evaluated (Zhai & Lafferty, 2001). The authors argued that smoothing actually has two roles: (i) improving the probability estimates of a document model, which is especially important for short documents, and (ii) "facilitating" the generation of common terms (a *tf.idf* like function). Dirichlet smoothing appears to be good for the former role, and linear interpolation (which is also called Jelinek-Mercer smoothing) is a good strategy for the latter function. In the experiments reported here, we have smoothed all generating models by

linear interpolation. We did some preliminary experiments with Dirichlet smoothing, but did not find significant improvements for the tracking task.

Linear interpolation based smoothing of e.g. a topic model is defined as follows:

$$P(w|T) = \lambda P(w|T) + (1-\lambda)P(w|C)) \tag{100}$$

The probability of sampling a term w from topic model $P(w|T)$ is estimated on the set of training stories using a maximum likelihood estimator. This estimate is interpolated with the marginal $P(w|C)$ which is computed on a large background corpus (the entire TDT2 corpus).

7.2.5. Parametric score normalization. We have seen in section 7.2.1 that it is easy to normalize generative probabilities for differences in length. Length normalized generative probabilities have a sound information theoretic interpretation. Length might not be the only topic dependent score dependency we we have to correct for. For example in a model which is based on the query cross-entropy $CER(Q;D,C)$ with smoothing based on linear interpolation, the median of the score distribution for each topic will differ, since it is directly correlated with the average specificity of topic terms. A brief look at a couple of extreme cases of title queries from the TREC ad hoc collection shows the following:

Query 403: osteoporosis: A query of a single very specific word will yield high scores for those documents that contain "osteoporosis". Since $P(w|D)$ is much higher than $P(w|C)$, the term weight is essentially determined by the ratio $\log(P(w|D)/P(w|C))$. Documents that do not contain the term "osteoporosis" do all receive the constant score $\log((1-\lambda))$ due to smoothing.

Query 410: Schengen agreement: This query consists of a quite specific proper name and the fairly general term "agreement". The contribution of "Schengen" to the total score of a document is much higher than "agreement". If a document does not contain "Schengen" it will not be relevant, therefore the score distributions between relevant documents are well separated[6].

Query 422: heroic acts: This query does not contain any rare terms, consequently document scores of relevant documents are lower.

Even though maximum likelihood procedures normalize for most of the length variations in topics for $CER(S;T,C)$ models, we still expect length dependencies in the scores because the generating models are smoothed. A longer topic will have a higher probability to have overlapping terms with stories than a shorter topic, which we expect to see in the score distribution.

The examples make clear that the score distribution of relevant documents (say the documents that contain most of the important terms) is dependent on the query. Queries formulated with mostly specific terms, will produce higher scores. The score distribution of non-relevant documents containing any of the query terms does also depend on the query. A perfect tracking system would produce separated distributions of relevant and non-relevant stories with equal medians and variances across topics, because of the single threshold. In reality, distributions are never perfectly separated (this would mean that $Precision = Recall = 1$). But we might be able to normalize scores distributions.

[6]This can clearly be seen in figure 7.10, which we will discuss later.

Score distributions have been studied by different researchers in the context of collection fusion (Baumgarten, 1997, 1999; Manmatha et al., 2001) or adaptive filtering (Arampatzis & Hameren, 2001). These researchers tried to model score distributions of relevant and non-relevant documents by fitting the observed data with parametric mixture models (e.g. Gaussian for relevant documents and exponential or Gamma for non-relevant documents). If the parametric models are a good fit of the data, it just suffices to estimate the model parameters to calculate the probability of relevance at each point in the mixture distribution. Unfortunately, we have very little training data for the distribution of the relevant documents in the case of tracking, so an approach like (Manmatha et al., 2001) is not feasible here. Instead, we could try to just estimate the parameters of the model for the non-relevant stories and assume that the concentration of relevant documents in the right tail of this distribution is high and hope that there is a more or less similar inverse relationship between the density of non-relevant and relevant stories in this area of the curve. This normalization strategy was proposed and evaluated for TDT tasks by researchers at BBN (Jin et al., 1999). They modeled the distribution of non-relevant documents by a Gaussian distribution, which can be justified by the central limit theorem for some of the models we have discussed. Indeed, the topic likelihood model score can be seen as a sum of independent random discrete variables. When a topic is long enough, the distribution can be approximated by the Gaussian distribution. It is unclear, whether this also holds for the story likelihood model, since the score is composed of a different number of variables for each story.

We implemented the Gaussian score normalization as follows: For each topic we calculated the scores of 5000 stories taken from the TDT Pilot corpus, we assumed these were non-relevant, since they predate the test topics.[7] We subsequently computed the mean and standard deviation of this set of scores. These distribution parameters were used to normalize the raw score τ in the following way:

$$\tau' = (\tau - \mu)/\sigma \tag{101}$$

7.3. Experiments

The generative models presented in the previous section will now be compared on two different test collections. Before presenting the actual data, the models will be briefly represented in section 7.3.1 followed by background information about the test collections and test metrics that we used in sections 7.3.2 and 7.3.3.

7.3.1. Experimental conditions. For our tracking experiments we plan to compare the following models:

story cross-entropy reduction: $CER(S;T,C)$*:* this is the model described in (92).

topic cross-entropy reduction: $CER(T;S,C)$*:* this is the model described in (96).

[7] Some of these stories could be considered relevant under a more liberal definition. Removal of these outliers has been reported to improve parameter estimation (Jin et al., 1999)

KL divergence: $KL(S||T)$ and $KL(T||S)$: recently, several researchers have argued that the KL-divergence can be viewed as a general model underlying generative probabilistic models for IR.

The first two models are motivated by the probability ranking principle. Query cross-entropy reduction is based on a model for ranking queries, but can also be used to rank documents. The KL divergence model is motivated as a loss function in a risk minimization framework, which does not explicitly model relevance.

Apart from comparing the effectiveness of the models as such, we will investigate the relative importance of several normalization components that are inherent to the models, namely the length normalization and the fact that the first two models compare entropy with respect to a common ground. We also evaluate the effectiveness of the Gaussian normalization and its interaction with different smoothing techniques.

7.3.2. The TDT evaluation method: DET curves. The TDT community has developed its own evaluation methodology, which is different from the evaluation of IR tasks that we discussed extensively in chapter 4. All of the TDT tasks are cast as detection tasks. In contrast to TREC experiments, the complete test set for each topic of interest is annotated for relevance. Tracking performance is characterized in terms of the probability of miss and false alarm errors ($P_{Miss} = P(\neg ret|target)$ and $P_{FA} = P(ret|\neg target)$). To speak in terms of the more established and well-known precision and recall measures: a low P_{Miss} corresponds to high recall, while a low P_{FA} corresponds to high precision. These error probabilities are combined into a single cost measure C_{Det}, by assigning costs to miss and false alarm errors (Fiscus & Doddington, 2002):

$$C_{Det} = C_{Miss} \cdot P_{Miss} \cdot P_{target} + C_{FA} \cdot P_{FA} \cdot P_{\neg target} \tag{102}$$

where C_{Miss} and C_{FA} are the costs of a miss and a false alarm respectively; P_{Miss} and P_{FA} are the conditional probabilities of a miss and a false alarm respectively; P_{target} and $P_{\neg target}$ are the a priori target probabilities ($P_{\neg target} = 1 - P_{target}$).

Then, C_{Det} is normalized so that $(C_{Det})_{Norm}$ cannot be less than one without extracting information from the source data:

$$(C_{Det})_{Norm} = \frac{C_{Det}}{min(C_{Miss} \cdot P_{target}, C_{FA} \cdot P_{\neg target})} \tag{103}$$

Thus the absolute value of $(C_{Det})_{Norm}$ is a direct measure of the relative cost of the TDT system (Doddington & Fiscus, 2002).

The error probability is estimated by accumulating errors separately for each topic and by taking the average of the error probabilities over topics, with equal weight assigned to each topic. The following parameters were determined a-priori: $C_{Miss} = 1$, $C_{FA} = 0.1$, and $P_{target} = 0.02$. The Detection Error Tradeoff (DET) curve is the equivalent of a precision-recall plot for ad hoc experiments. The DET plot shows what happens when the decision threshold of the tracking system performs a sweep from an (infinitely) high value to an (infinitely) low value. Obviously, at the beginning of the parameter sweep, the system will have zero false alarms but will not detect any relevant stories either and moves to the opposite end of the trade-off spectrum when the threshold is

decreased. An example DET plot is figure 7.1. A good curve in a DET plot is a relatively straight curve with a negative slope. The steeper the curve, the better.

Note that the DET curves produced by the TDT evaluation software have *normal deviant* scales in order to magnify the high performance area of the curve. A straight line indicates that the underlying error distributions of P_{Miss} and P_{FA} are normal.

7.3.3. Description of the test collections. Currently, the Linguistic Data Consortium (LDC) has three corpora available to support TDT research[8] (Cieri et al., 2000). The TDT-Pilot corpus contains newswire and transcripts of news broadcasts, all in English, and is annotated for 25 news events. The TDT2 and TDT3 corpora are multilingual (TDT2: Chinese and English, TDT3: Chinese, English, and Arabic) and contain both audio and text. In addition ASR transcriptions and closed captions of the audio data as well as automatic translations of the non-English data are provided. TDT2 and TDT3 are annotated for 100 and 120 news events respectively.

For the experiments in this chapter, we used a subset of the TDT2 corpus consisting of the topics of the months May and June (17 topics). Our study is limited to a simplified dataset: the ASR transcripts of the audio sources and the MT processed foreign data sources. We will not study source specific dependencies, i.e. the dataset is regarded as a uniform and monolingual collection of news stories. All experiments were done with just one training story per topic.

Because experimentation with tracking is a time consuming process, we also simulated a tracking task by using TREC ad hoc runs. Such an experiment was suggested by Ng (Ng, 2000a) who simulated a binary classification task on TREC ad hoc data with a fixed threshold. We will discuss further details in section 7.3.5.

7.3.4. Experiments on TDT test collection. We will first present a comparison of the basic models which have $P(S|T)$ as their core element: topic likelihood models. All experiments are based on smoothing by linear interpolation with a fixed $\lambda = 0.85$.

Figure 7.1 shows the results of several variant models in a DET curve. The basic story likelihood model $P(S|T)$ is hardly better than a random system (with a constant $P_r(n)$). This is not surprising, since the likelihood is not normalized. The relative effect of the two normalization components, i.e. normalizing by the a-priori story likelihood and story length normalization, is quite different. Taking the likelihood ratio is the fundamental step, which realizes the *idf*-like term weighting and converts likelihood ranking to log-odds ranking (cf. formula (91)). Story length normalization removes some variance in the scores due to length differences and improves upon the LLR model for most threshold levels. The tracking model ($CER(S;C;T)$) combines both normalization steps.

Surprisingly, the performance of $KL(S||T)$ is even worse than the length normalized likelihood $H(S,T)$ (=$1/|S|log(P(S|T))$. The KL- divergence can be seen as an entropy normalized version of the latter: $KL(S||T) = -H(S,T)+H(S)$, whereas the cross-entropy reduction ranking formula normalizes by adding the cross-entropy with the background collection: $CER(S;T,C) = -H(S,T) + H(S,C)$. The experimental results make clear that normalizing with entropy deteriorates results, whereas normalizing with $P(S|C)$ (or its length normalized version $H(S,C)$) is an essential step in achieving good results.

[8] http://www.ldc.upenn.edu/Projects/TDT

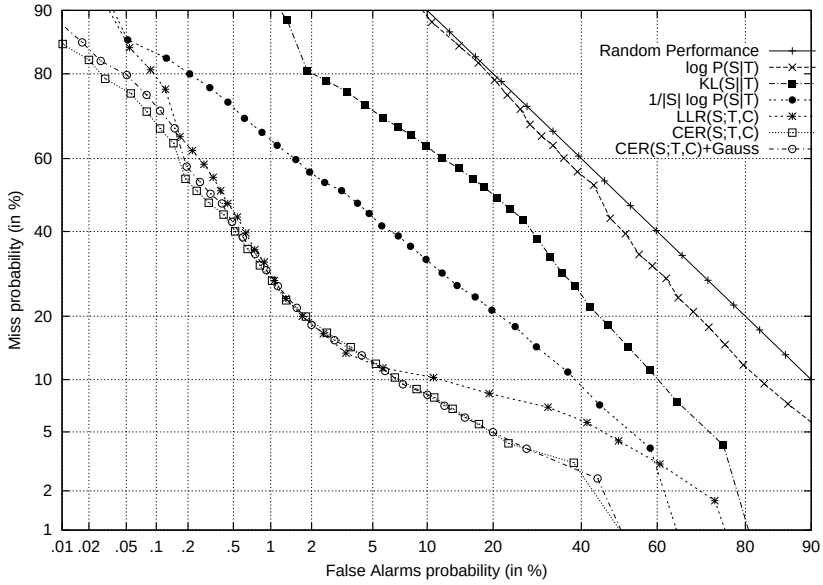

Figure 7.1. Comparison of different tracking models based on $P(S|T)$.

We repeated the same experiments for the reversed direction: with the topics generated by the stories. Results are plotted in figure 7.2. The relative performance of the $P(T|S)$ based variant models is roughly equivalent to the variants of the $P(S|T)$ models, with the exception of the models which are not based on a likelihood ratio. Again, the main performance improvement is achieved by normalizing $P(T|S)$ with the prior likelihood $P(T|C)$, which is equivalent to ranking by log-odds of being liked. Length normalization improves performance at both ends of the DET-plot and results in a straighter curve. The length normalized likelihood model $1/|T| \log P(T|S)$ performs worse than its reverse counterpart. This is due to the fact that scores are not normalized for average term specificity across topics. An even more striking phenomenon is the step-like behaviour of the unnormalized $P(T|S)$. This is due to the fact that the score distributions of plain $P(T|S)$ are linearly dependent on topic lengths and consequently their medians are located quite far apart. We will illustrate this effect by some boxplots.

A boxplot is a graphical summary of a distribution, showing its median, dispersion and skewness. Boxplots are extremely helpful to compare different distributions. A boxplot is defined by five datapoints: the smallest value, the first quartile, the median, the third quartile and the largest value. The area between the first and third quartile (interquartile range) is depicted by a box, with a line marking the median. (figure 7.5 is a good example.) The boxplots in this chapter also have whiskers that mark either the

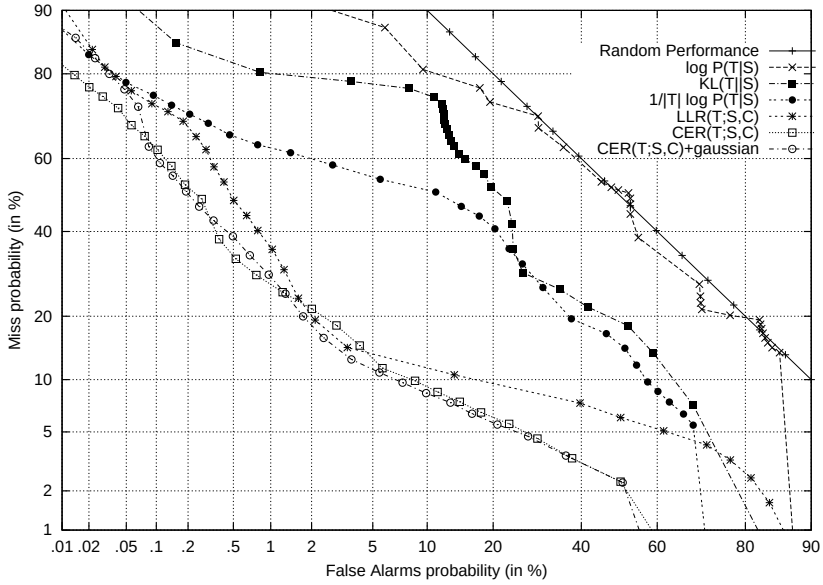

Figure 7.2. Comparison of different tracking models based on $P(T|S)$.

smallest or largest value or the area that extends 1.5 times the interquartile range from the first or third quartile.

Figures 7.3 and 7.4 show boxplots of the distributions of $CER(T;S,C)$ and $P(T|S)$ respectively. The first plot shows that the bodies of the distributions for all topics are quite well aligned. The distributions are skewed and have a long right tail, because they are in fact mixtures of a large set of relevant stories and a small set of non-relevant stories with a higher median. Figure 7.4 gives an explanation why the DET plot curve of this model is so wobbly: the distributions of the individual topics do not even overlap for a few cases: sweeping the evaluation threshold will bring in the stories of each topic as separate blocks. This means that the probability of relevance will increase and decrease locally as we decrease the threshold, causing convex and concave segments (cf. section 7.3.2). Because the boxes are hardly visible in both cases, we show an example of a more dispersed distribution: $KL(T||S)$ in figure 7.5. The fact that the distributions lack a long right tail is a sign that relevant and non-relevant documents are probably not well separated. Finally, an example of well-aligned symmetrical distributions is $LLR(S;T,C)$ in figure 7.6. The symmetry is due to the fact that scores are not length normalized, long stories that do not have word overlap with the topic will have high negative scores, long stories with good word overlap with the topic will have high positive scores. Figure 7.7 shows that indeed there is some topic length effect for the $CER(S;T,C)$ model as we hypothesized in section 7.2.5. For example, the first topic has length 395 and the second

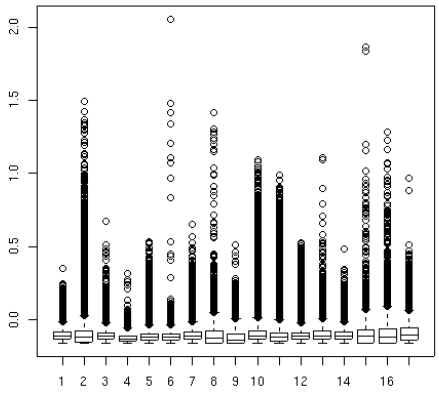

Figure 7.3. Score distributions of $CER(T;S,C)$

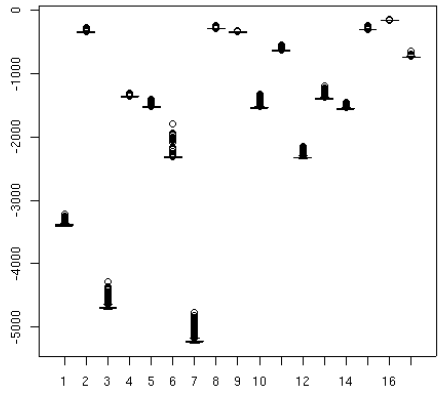

Figure 7.4. Score distributions of $P(T|S)$

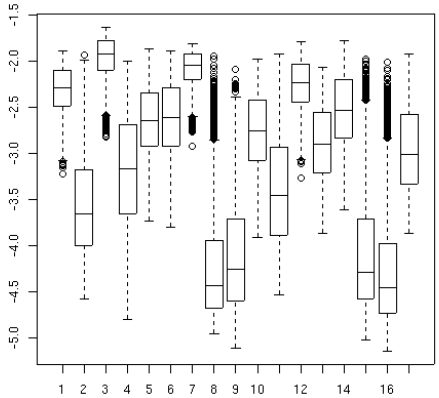

Figure 7.5. Score distributions of $KL(T||S)$

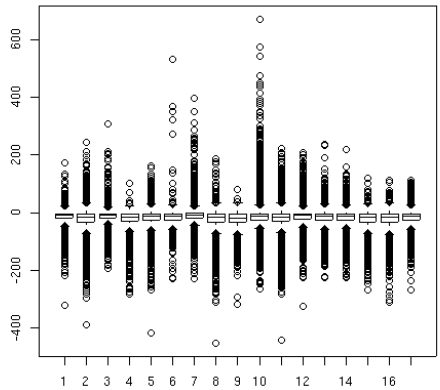

Figure 7.6. Score distributions of $LLR(S;T,C)$

has length 43, which results in lower scores for the bulk of the distribution. Figure 7.8 shows score distributions of the same model after applying Gaussian normalization. Indeed the boxes are better aligned, but differences are small. The normalization resulted however in some performance loss in the high precision area, cf. figure 7.1. We have also applied Gaussian normalization to the $LLR(S;T,C)$ model, which is not normalized for story length. In this case, the Gaussian normalization deteriorated results, even though medians were well aligned. We think that this is due to the fact that the variance in the

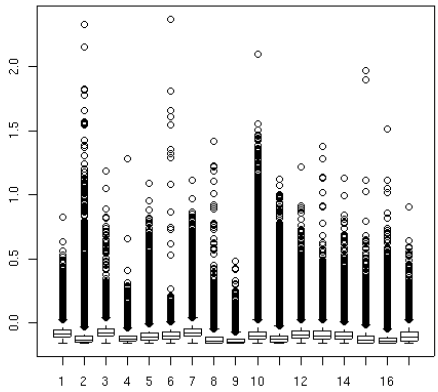

Figure 7.7. Score distributions of $CER(S;T,C)$

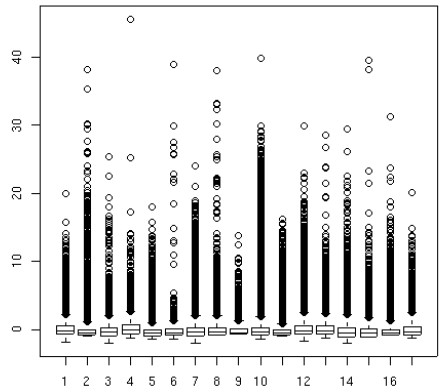

Figure 7.8. $CER(S;T,C)$ + Gaussian normalization

score distribution is due to differences in length, which can be normalized in a more effective way. Gaussian normalization of the model in the reverse direction: $CER(T;S,C)$ had similar effects: a small performance loss in the high precision area and no effect for the high recall area. Further investigation is needed in order to understand why the Gaussian normalization is not effective. There are several possibilities: (i) scores are already quite well normalized, (ii) the score distribution differs too much from the normal distribution, or (iii) outliers hurt the estimation of the distribution parameters.

Since both directions of the cross-entropy reduction ranking function (CER) work well, there might be some potential for improvement by combining both models. We did some initial experiments which were based on a simple score averaging procedure. A side effect of this method is that scores become symmetric. It is exactly this symmetrical model that has proven to be effective for the TDT detection task (Spitters & Kraaij, 2002). The resulting system performed worse than each of the individual components, but after applying Gaussian normalization the system was a little bit more effective than a model based on just a single direction. Further research is needed to find an optimal combination/normalization procedure.

7.3.5. Simulating tracking on TREC ad hoc data. We complemented the runs on the TDT2 corpus with experiments on TREC ad hoc data. The main reason is that most of the data was available already, and provided a rich resource for research on score normalization. Since ad hoc runs output a list of scored documents, we could simulate a tracking system based on the cross-entropy reduction ranking formula, by placing a threshold. We applied two methods to implement this idea. The first method is based on `trec_eval`, the second on the TDT evaluation software `TDT3eval`.

The basic idea is to evaluate all 50 topics of an ad hoc run by a single threshold. Standard `trec_eval` does not support this kind of evaluation. However, it can be simulated

by replacing all topic-id's in both the runs and the qrel file by a single topic-id. Of course, this evaluation is different from the TDT evaluation, since this method does not involve topic averaging, so topics with many relevant documents will dominate the results. Still, this evaluation is a quick and easy method to assess score stability across topics when TDT evaluation software is not available. We tested this method on the TREC-8 ad hoc test collection, for both title and full queries. Table 7.2 shows the results of our experi-

run name	title (tracking)	title (ad hoc)	full (tracking)	full (ad hoc)
$P(Q\|D)$	0.0874	0.2322	0.1358	0.2724
$LLR(Q;D,C)$	0.1334	0.2321	0.1577	0.2723
$CER(Q;D,C)$	0.1294	0.2324	0.1581	0.2723
$KL(Q\|\|D)$	0.0845	0.2322	0.1356	0.2723

Table 7.2. Tracking simulation on TREC-8 Ad Hoc collection (mean average precision)

ments, using four weighting schemes: straight (log) query likelihood, log-likelihood ratio, normalized log-likelihood ratio and KL. We see that the influence of the particular normalization strategy is quite strong on the tracking task, while - as was expected - there is no influence on the ad hoc task. Indeed the normalization strategies just add topic specific constants, which do not influence the ad hoc results. There seems to be no big difference between LLR and CER, but that might be due to the averaging strategy, which is not weighted across topics. CER is a bit less effective than LLR for title queries, but that can be explained by the difference in query term specificity for short (1-3 word) queries. A single word TREC title query must be very specific (e.g. topic 403: "osteoporosis") in order to be effective. Two and three word queries often use less specific words and thus their scores will be lower with CER, which is normalized for query length. Still two or three word queries can be just as effective as one word queries, so there is no reason to down-normalize their scores. This effect was confirmed by the boxplots for these runs, shown in figures 7.9 and 7.10. The title queries with the highest CER scores (403 and 424) are single word queries. The boxplots show a mix of topics to visualize the topic normalization, the score distributions of the first 25 topics (topic 401-425) are based on title queries, the rightmost 25 distributions are based on the full queries (topic 426-450).

The KL-divergence based run really performs disappointingly. We can conclude that KL as such is not a suitable model for tracking, at least not for models estimated with maximum likelihood estimation. We also ran the TDT evaluation scripts on the TREC data after applying a conversion step. The difference with the previous method, is that the TDT evaluation procedure averages P_{FA} and P_{Miss} across topics. The results of the run based on the full topics are shown in plot 7.11. The best performance is reached by CER, which is just a bit better than LLR. Again KL yields a very disappointing result.

7.4. Discussion

One of the main challenges of designing a tracking system is to normalize scores across topics. Since topics are of a very different nature, and there is no direct relationship

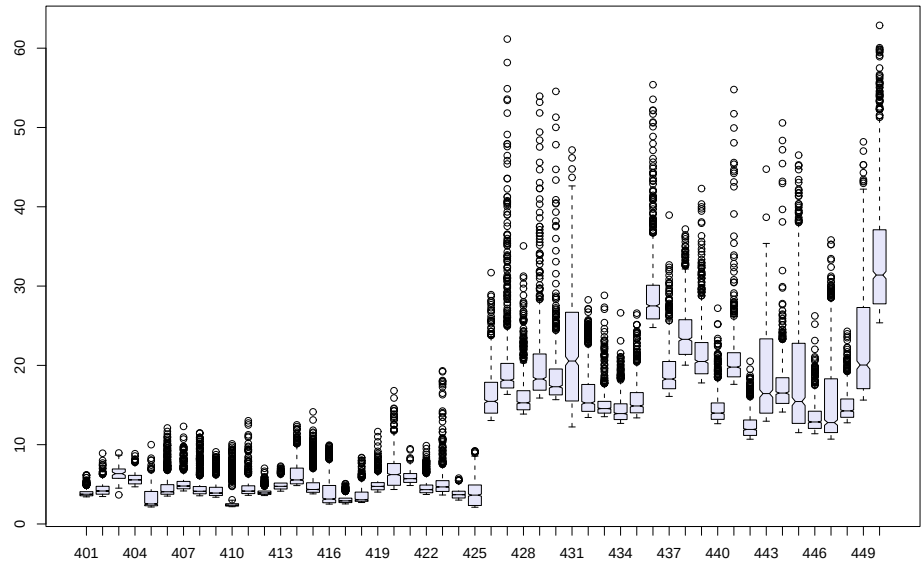

Figure 7.9. Tracking simulation: LLR

between the score distribution of the models and probability of relevance, this is quite a hard task. An ideal system would produce the probability of relevance of each test story/document as a score. A system could then be optimized for a certain cost or utility function, by setting a threshold on a certain probability of relevance value. However, there is only an indirect relationship between score distribution and probability of relevance. We have seen that scores can be dependent on story and/or topic length and on the term specificity of their formulation. Some topics are "easy" i.e. the score distributions of relevant and irrelevant stories are well separated. We have tried to cope with these differences using different techniques, (i) we used a model with inherent normalization: the (normalized) log likelihood ratio (ii) we tried to model the score distributions themselves.

The log-likelihood ratio based tracking models are directly derived from the probability of relevance and thus have the advantage that the scores have a clear interpretation and a common reference point. They compare the generative probability of a story given the topic (or vice versa) in comparison with the a-priori probability. Likelihood ratios are in fact a form of statistical hypothesis tests, where each generative model is one hypothesis. Previously reported results of our CER based system for the official TDT 2000 tracking task were competitive (Spitters & Kraaij, 2001). We therefore conclude that language models can form the basis for an effective tracking system indeed, provided that the models are properly normalized. Our experiments with TDT and TREC data showed

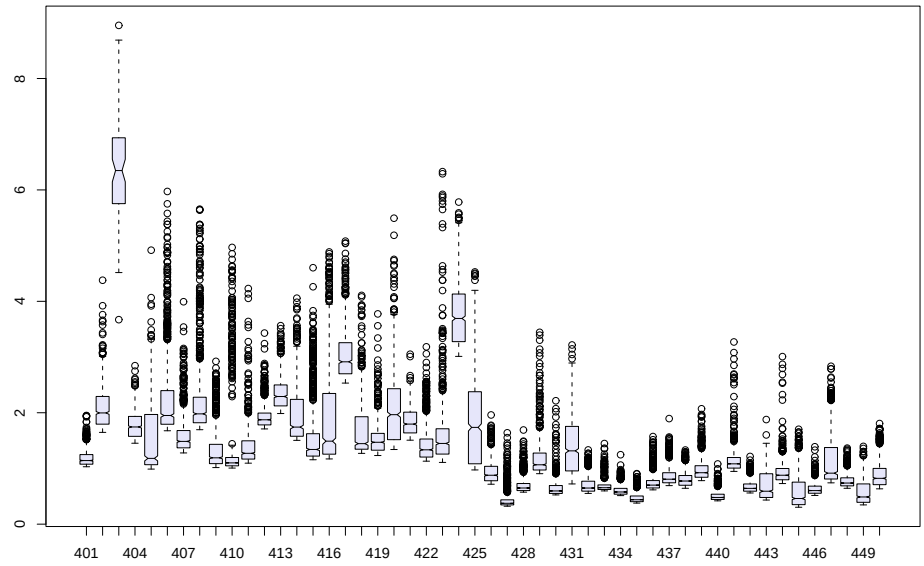

Figure 7.10. Tracking simulation: CER

that normalization using an a-priori model is a key point. But the function of the normalizing likelihood in the denominator is different for the two directions of the model. In the story likelihood case, the $P(S|C)$ component normalizes scores for across story differences in term specificity, whereas in the topic likelihood case, the $P(T|C)$ component normalizes scores for across topic differences in term specificity. Scores can be normalized further by applying length normalization. Both directions of the model have comparable tracking effectiveness for the case of a single training document.

We also evaluated a method for score normalization which tries to fit the distribution of non-relevant stories by a Gaussian distribution. This normalization was not really effective for the $CER(S;T,C)$ model and even seriously hurt the effectiveness of the $LLR(S;T,C)$ model. We think that the $LLR(S;T,C)$ model is not suitable for Gaussian normalization since the score variance is dominated by differences in story length, which should be removed prior to Gaussian normalization. BBN has reported favourable results with Gaussian normalization (Leek et al., 1999). We conjecture that Gaussian normalization could work for their 'IR' tracking model, which is equivalent to $P(T|S)$: the straight topic likelihood. Gaussian normalization is able to normalize across topic differences. However, a simpler method is to work with the likelihood ratio $P(T|S)/P(S)$ instead. After all, unlike ad hoc the denominator $P(S)$ is not a constant.

Despite the intuitive appeal of KL - measuring the dissimilarity between distributions - our experiments with KL for tracking yielded disappointing results for both directions $KL(S||T)$ and $KL(T||S)$. The KL-divergence has usually been proposed in an ad hoc

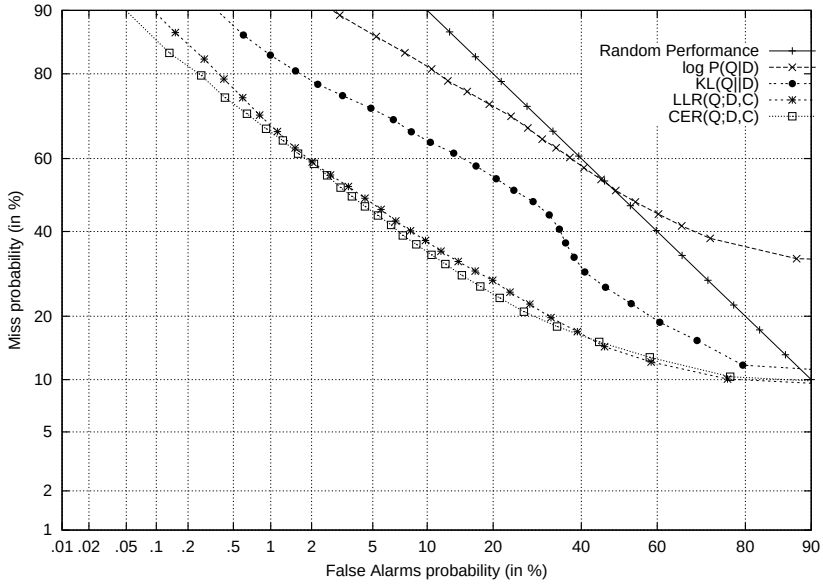

Figure 7.11. DET plot of tracking simulation on the TREC 8 ad hoc full topics

query-likelihood context. In that case KL reduces to pure query-likelihood, since the normalizing entropy $H(Q)$ in the KL divergence can be discarded because it is a constant. This cannot be done in a tracking context and we have seen that normalizing a cross-entropy by its entropy is not effective in a tracking context. We have also shown that normalizing by the prior probability of a topic as measured by its cross-entropy with the collection model is effective. Informally we could say that the KL divergence based scores are not properly normalized. The problem of using KL divergence for tracking is that the scores lack a common point of reference. Dissimilarity is measured w.r.t. different reference distributions and since KL is not a metric, these scores cannot be compared. A more formal criticism on the use of KL divergence for tracking is that KL based models lack the notion of relevance. We have shown however, that both directions of the normalized log likelihood ratio, which are direct derivations of probability of relevance based ranking, are quite effective for tracking.

This analysis has been recently confirmed by independent research in the area of story link detection (Lavrenko et al., 2002a). Lavrenko found that pure KL is not effective for story link detection and proposed the so-called "Clarity-adjusted KL" topic similarity measure to correct for the fact that KL does not concentrate on informative (in the *idf* sense) terms when computing the (dis)similarity score. This adjusted KL measure is defined as $-KL(T||S) + \text{Clarity}(T)$, where clarity is defined as $KL(T||C)$ (Cronen-Townsend et al., 2002). Indeed, when comparing this definition to formula (99), the Clarity-adjusted

KL divergence is equivalent[9] to the cross-entropy reduction. The CER similarity measure can thus be motivated by two frameworks: (i) direct derivation from the log-odds of relevance and (ii) clarity adjusted version of KL divergence.

We also evaluated the query-likelihood models by a simulation of the tracking task on the TREC-8 ad hoc collection. We know that there is a real difference between ad hoc topics and TDT topics. This difference is one of the reasons that score normalization effectiveness differs across these tasks. TDT topics are just stories describing a particular event. Ad hoc topics are structured queries which are stated in a particular jargon. The bag-of-words approach we took for ad hoc query construction showed a clearly visible difference between title queries and full queries. Even our best normalization strategy (CER) could not "smooth out" the differences in score distributions between these two types of queries. We plan to develop topic-type (e.g. title versus full) specific query distribution estimation methods, which we hope will enable us to further normalize scores.

7.5. Conclusions

Our aim was to select and adjust generative probabilistic models that work well for both the ad hoc task and the tracking task, because a tracking system puts just one additional constraint on matching function: across topic comparability of scores, which does not influence ranking for the ad hoc task. With the probability ranking principle as a starting point, we reviewed two lines of probabilistic modeling, either based on the document likelihood ratio or the query likelihood ratio. We evaluated variants of both models, based on length normalization and Gaussian normalization. We found that both directions of the log-likelihood ratio work well. The essential normalization component in the CER model is the a-priori likelihood (or cross-entropy) of the generated text in the denominator. Effectiveness can be further enhanced by length normalization.

We have not been able to show performance increase by Gaussian normalization. The CER model is related to the negated KL divergence since both measures are based on the cross-entropy. We found that KL divergence is not an effective scoring function for tracking, because the scores are not comparable across topics (for $KL(T||S)$) or across stories (for $KL(S||T)$). The principal reason seems to be the fact that the application of KL divergence as a similarity measure for the tracking task lacks normalization with respect to a common reference distribution. We also claim that the CER model (both directions) has a stronger theoretical foundation, since it is directly derived from the log-odds of relevance.

[9] Note however that in Lavrenko's framework, topic models are estimated using the relevance model technique.

CHAPTER 8

Summary and conclusions

In this thesis we have explored several ways to adapt an IR system based on the language modeling framework for specific IR tasks. In particular, we studied cross-language information retrieval, morphological normalisation for Dutch ad hoc retrieval and topic tracking. These studies were motivated by three main research questions:

(1) How can linguistic resources be optimally embedded into LM-based IR models?
(2) Is it possible to give a single formulation of a document ranking function based on generative probabilistic models, which can be applied for various specific IR tasks: cross-language information retrieval, monolingual ad hoc retrieval and topic tracking?
(3) Is it possible to define improved guidelines for the statistical validation of IR experiments?

We have approached the first two questions with a combination of model design and model evaluation. The third research question was motivated by the need to have a methodological basis for the experiments that we carried out to validate hypotheses in relation to the first two research questions. For the third research question we carried out a literature review and validated assumptions of various standard tests. Since no equivocal opinion emerged from our investigations, the guidelines proposed for the validation of IR research were based on the general principles of being selective in testing hypotheses and conservative in drawing conclusions. In the following sections, we will summarize our conclusions for each of the research questions and finish with a section on possible future work.

8.1. The optimal embedding of linguistic resources in LM-based IR models

Our main research question concerned the optimal way to combine linguistic resources with IR systems based on generative probabilistic models. In chapter 5, we discussed this problem in the context of cross-language information retrieval (CLIR), in chapter 6, we evaluated several variations on language modeling for ad hoc retrieval with integrated morphological normalization. The main conclusions will be summarized in the following paragraphs.

Alternative probabilistic CLIR models. We explored several ways to embed simple word-by-word translation into a cross-entropy reduction based IR model, yielding several variant CLIR models. The main variants are (i) to map the document language model into the

query language, (ii) to map the query language model into the document language or (iii) to map both the document and query language model into a third language. Such models do not require explicit disambiguation before translation, — translation uncertainty is encoded in a probability distribution of translation alternatives —, and use the target documents themselves as the main disambiguation resource using the co-ordinating power of the retrieval model. Although such an approach is very simplistic — only global estimates about the probabilities of certain senses are available — the method has proven quite effective for the medium length queries we studied. The probabilistic CLIR model variants yielded better performance than the combination of machine translation and monolingual IR. A CLIR run with combined translation resources yielded performance levels close to a monolingual run. The method is also very efficient, since no time consuming query or document-specific processing (e.g., disambiguation, EM-algorithm or Viterbi search) is necessary.

Factors determining CLIR effectiveness. Determining factors for CLIR effectiveness in order of importance are: (i) coverage of translation dictionaries, (ii) weighting of translation alternatives, (iii) number of translation alternatives. It is crucial to have at least one good translation for each query term and it is also important that the weighting of translation alternatives is based on good probability estimates.

Interaction between translation resource type and CLIR model type. Parallel corpora and lexical databases behave quite differently when they are used as a translation resource for the probabilistic CLIR models. If large enough, the former have the potential to be far more effective than bilingual dictionaries. This is mainly due to better translation probability estimates and the availability of many proper name translations and expansion terms. Probabilistic CLIR models based on translation models mined from parallel corpora have a significantly better performance than the popular CLIR model based on treating translation alternatives as synonyms. The synonym-based approach clearly breaks down when many translation alternatives have to be dealt with, whereas the probabilistic models make an effective use of the weighted translation alternatives.

Since parallel corpora mined from the Web are noisy, it is important to clean the generated translation models by applying effective pruning methods. This turned out to be the case especially for the CLIR model that maps document models into the query language, since documents contain many more rare terms than queries. The translations of rare terms are not reliable, due to data sparseness.

Ambiguity. Previous work on language modeling for CLIR only investigated the variant where the document language model is translated in to the query language. We have found that translation in the reverse direction is also effective (although we only tested medium length queries). Though the former approach has a strong coordination effect, which helps to resolve ambiguity by enforcing compatible meanings, the latter model has the additional advantage of getting better leverage from expansion terms. We have strong indications that ambiguity does not play a dominant role in the query collections used in CLEF and that the detrimental effect of unresolved ambiguity on the effectiveness of some short queries is more than compensated by the benefit of good expansion terms on the majority of the queries.

The importance of accurate and robust estimation. There are some indications that estimation quality plays a significant role in the relative performance of the variant probabilistic CLIR models. It is well known that the main reason why "generating the query by the document" is more effective than "generating the document by the query" is the fact that there is far more data available to estimate a document language model than for the estimation of a query language model (Greiff & Morgan, 2003). The performance difference between both model directions disappears when topics and documents have approximately the same length, as shown in chapter 7, for the topic tracking experiments. For the CLIR task we have only evaluated models, for which the cross-entropy is measured between the query language model and the document model, and for which the document language model is the generating distribution. We hypothesized that, since there is more data available to map a document model into the query language than vice versa, a CLIR model based on the former would be more effective than the latter. However, experimental results were not conclusive on this issue. One of the problems is that it is not possible to compare results across different document collections. There is however some empirical evidence that the direction in which the language models are trained correlates with translation accuracy. The CLIR models that use translations trained on word alignment models that align from a more verbose to a less verbose language (e.g., French,Italian→English) are (with one exception) more effective than the CLIR models that use the reverse translation direction. It seems thus beneficial to apply matching (by measuring cross-entropy) in a reduced feature-space, since this will lead to more robust probability estimates.

Reduced feature-space. The improved effectiveness of matching in a reduced feature-space was confirmed by our extensive experiments in chapter 6. We evaluated different conflation architectures, based on matching in the feature-space of wordforms vs. matching in the feature-space of stems. Despite indications that more refined weighting techniques for conflation terms could increase retrieval effectiveness in comparison with standard conflation, it was found that these more sophisticated techniques did not yield significant improvements. In particular, we evaluated several language models with an integrated weighted stemming component, motivated on linguistic, heuristic or corpus-based grounds. We must conclude that the strategy to replace all conflation terms by one stem and match in the reduced feature-space of stems is effective, most probably because there is not enough data for reliable estimation in the feature space of wordforms.

Morphological normalization techniques. In addition, we evaluated various techniques for morphological normalization for Dutch monolingual IR. We did not find significant differences in retrieval performance for different affix removal methods. We tested full morphological analysis using a dictionary (CELEX), a Dutch version of the Porter algorithm and conflation based on fuzzy matching. All three methods significantly improved mean average precision on the UPLIFT collection; the best result (+15% m.a.p.) was produced by dictionary-based normalization. Further improvements of retrieval effectiveness can be achieved by splitting compounds and adding both the original compound and its components as index terms. Finally, we evaluated the effectiveness of stemming

at the individual query level and found that (contrary to what is often assumed) stemming does improve both recall and precision.

8.2. A single ranking model for different IR tasks

Throughout our experiments with different tasks we have experienced that the language modeling framework is a quite versatile and robust formalism for modeling different variations of IR tasks. At the outset of our work, several different variants of language modeling for IR were published for ad hoc tasks (cf. section 2.6.3) and topic tracking (section 7.2). We hypothesized that it should be possible to construct a LM-based document ranking formula that would support both the ad hoc and tracking task. The topic tracking task is not unlike the ad hoc search task (both tasks contain a ranking component) but the topic tracking task requires comparable scores across topics. We have shown that our formulation based on cross-entropy reduction with respect to a common background model, which is a reformulation of the model by Ng (2000a) and closely related to Hiemstra (1998) and Miller et al. (1999b) fits our needs. This was demonstrated by a qualitative analysis of the score distribution properties of two popular LM-based topic tracking models with a different direction of generation. The qualitative analysis was empirically confirmed by comparing score distributions and performance with other systems on the topic tracking task. The Kullback-Leibler divergence has been proven an unsuitable model for topic tracking. We believe that it is also a wrong formalisation of the ad hoc retrieval task, contrary to what is claimed by the developers of the LEMUR research IR toolkit. Independently, other researchers (Lavrenko et al., 2002a) have proposed the so-called "clarity adjusted KL measure", instead of the KL measure for the normalized ranking score between two document models. As the clarity adjusted KL formula can be reduced to the cross-entropy reduction formula, the latter provides s a more concise formalization. The cross-entropy based formulation has the additional advantage that it allows for a transparent modeling of different translation directions for CLIR, since the cross-entropy reduction formula describes a metric between two language models, normalized by a third language model. Each language model can be estimated in many different ways, leveraging different types of knowledge resources e.g., translations or morphological knowledge.

8.3. Guidelines for statistical validation of IR experiments

We have reviewed several statistical significance tests for the comparison of pairs of system results or larger sets of results. A safe approach is to use non-parametric tests like the sign test or the Friedman test, but these methods have several disadvantages: (i) they are less sensitive, (ii) they do not provide confidence intervals that can be interpreted on the original measurement scale, and (iii) (for Friedman) are sensitive to the composition of the set of runs that are subjected to the comparison. We formulated several guidelines to overcome the latter problem. We also reviewed parametric methods and found that in many cases their assumptions about error distributions are not met. Our findings justify the recommendation to run several types of tests and only draw firm conclusions when tests have uniform results. In the same vain, the reliability of conclusions can be improved when experiments are carried out on multiple collections.

8.4. Future work

The work presented in this thesis has partially answered the research questions that formed the start of our investigations. Fortunately, not all problems have been solved. We will enumerate a few of the more promising areas for continued and further research that we have identified in the "slipstream" of the research reported.

translation vs. expansion: Our experiments indicate that a large part of the success of using corpus-based statistical translation dictionaries is due to the fact that they bring in a form of expansion with related terms. Further experiments are needed to determine whether the hidden semantic relationships in a parallel corpus can be exploited in more effective ways for either cross-lingual or monolingual retrieval, e.g., by looking at sentence alignment instead of, or in addition to word alignment.

structured vs. unstructured queries: A related question is whether it is possible to construct automatic procedures to build structured queries that optimally handle the different types of statistical translation terms. We presume that translation alternatives are best handled by a structured query operator, but that expansion terms are better handled as real expansion terms i.e. not under the scope of a structured query operator.

optimal reduced feature space: The previous issue relates to the more general question of defining an optimal feature space for measuring the cross-entropy. Our experiments in chapter 6 clearly showed that reducing the feature space is beneficial, however techniques that pursue this idea in a more principled way like LSI have their disadvantages. One area to explore is to make reduced feature spaces query specific in combination with more refined methods for query model estimation leveraging co-occurrence information from corpora.

PART III

Appendix, Bibliography and Index

APPENDIX A

SMART term weighting scheme codes

Salton and Buckley have developed a shorthand notation scheme to refer to the different term weighting schemes that they evaluated within the SMART project. Table A.1 lists the term weighting formulas which were compared in (Salton & Buckley, 1988). Unfortunately this encoding, which we will refer to as the IP&M[1] encoding, never got hold in subsequent publications. Instead, papers which use the SMART IR system use the smart-internal encoding which has never been published (Buckley, 1999). This encoding has a different semantics for the letter *n*, thus giving rise to confusion. We present both encodings and their motivation for the convenience of the reader, where the IP&M encoding is shown between parentheses.

The shorthand signatures consist of six letters. The first group of three letters refers to the term weighting which is applied to the document terms, the second group refers to the query term weighting. The first letter of each triple refers to the term frequency weighting component, the second to the collection frequency weighting component and the third letter to the normalization component. In principle, every term weighting scheme in the vector space tradition can be expressed signature consisting of a pair of triplets. A document ranking formula can be built from each possible combination of these six letters as follows: the document term weight is composed of a multiplication of the term weight (first letter) and the collection frequency weight (second letter). The resulting weight is subsequently normalized by the normalization component. The same procedure is applied for the computation of query term weights, but this time with the last three letters. The document score is finally obtained by taking the inner product of the weighted query and document vector.

[1] IP&M is a a shorthand for the journal *Information Processing and Management.*

code	term weighting component	explanation/motivation
Term frequency weight		
b (b)	1	Binary weight equal to 1 for terms present in the document.
n (t)	tf	Raw term frequency.
a (n)	$0.5 + 0.5\frac{tf}{\max(tf)}$	Augmented normalized term frequency.
l	$1 + \log(tf)$	Limit the effect of high frequency query terms.
L	$\frac{1+\log(tf)}{1+\log(\sum_{i=1}^{L} tf_i/L)}$	Down-weighting the effect of frequent terms in long documents by normalizing on average term frequency (L is size of indexing vocabulary).
d	$1 + \log(1 + \log(tf))$	Down-weight the effect of high frequency terms by a double logarithm.
Collection frequency weight		
n (x)	1	No collection frequency weighting.
t (f)	$\log \frac{N}{n}$	Multiply by inverse document frequency (*idf*), (N is number of documents in the collection, n is the number of documents to which a term is assigned).
p (p)	$\log \frac{N-n}{n}$	Probabilistic version of the inverse document frequency (cf. section 2.6.1).
Normalization component		
n (x)	1.0	No normalization.
c (c)	$1/\sqrt{\sum_{vector} w_i^2}$	Cosine normalization.
u	$\frac{1}{(1-s)p+s\cdot V_d}$	Pivoted unique normalization, where p is the pivot, s is the slope and V_d is the number of unique terms in the document.
b	$\frac{1}{0.8+0.2\cdot\frac{b_i}{\sum_{i=1}^{N} b_i/N}}$	Pivoted byte length normalization, where 0.8 is the pivot, 0.2 is the slope and b_i is the length of document i in bytes.

Table A.1. Term weighting components: The left column shows the SMART code scheme of the term weighting component with the corresponding IP&M code scheme between brackets (if applicable). Note that the term frequency weights only apply when $tf > 0$.

APPENDIX B

Okapi model components

Table B.1 gives an overview of the main Okapi weighting functions. The formulas consist of a term weighting function which is used to weigh each term in the query and in a global component which will be added to the RSV independently of the query terms. Experiments showed that the BM11 formula performed consistently better than BM1 on

code	term weight	global	explanation
BM0	1	–	Coordination level matching
BM1	$\log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_q}{k_3+tf_q}$	–	Robertson/Sparck Jones weights plus query term reweighting
BM15	$s_1 s_3 \cdot \frac{tf}{k_1+tf} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_q}{k_3+tf_q}$	$k_2 \cdot \lvert Q \rvert \frac{\Delta-d}{\Delta+d}$	BM1 plus within-document term frequency correction and document length correction
BM11	$s_1 s_3 \cdot \frac{tf}{\frac{k_1 \times d}{\Delta}+tf} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_q}{k_3+tf_q}$	$k_2 \cdot \lvert Q \rvert \frac{\Delta-d}{\Delta+d}$	BM15 plus within-document term frequency normalization by document length
BM25	$s_1 s_3 \cdot \frac{tf^{'}}{K^c+tf^{'}} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{tf_q}{k_3+tf_q}$	$k_2 \cdot \lvert Q \rvert \frac{\Delta-d}{\Delta+d}$	Combination of BM11 and BM15, where $K = k_1((1-b)+b\frac{d}{\Delta})$

Table B.1. Okapi term weighting functions

short and especially on long queries from the TREC-2 collection. The improved performance is mostly due to the heterogeneous document lengths of the TREC-2 collection which are better modeled by BM11. TREC-3 experiments finally resulted in the BM25 term weighting scheme (also known as the Okapi formula), which combined both BM11

and BM15 in one single formula, however, introducing two new constants b and c. The motivation for the new *tf* component of BM25 was that the *tf* component in BM11 is based on the "verbosity hypothesis" which might not always hold in the heterogeneous TREC collection. With $c > 1$ the new *tf* formula exhibits an s-shape similar to the 2-Poisson model. However, in practice, $c = 1$ was used, a higher order power did not help. In this ($c = 1$) case BM25 reduces to BM11 for $b = 1$ and to BM15 for $b = 0$. The BM25 weighting scheme performed only slightly better than BM11 in TREC-3. However, BM25 became very popular among TREC participants in later issues of TREC.

APPENDIX C

UPLIFT test collection

This appendix provides some documentation about the test collection that was developed during the UPLIFT project[1] (Kraaij & Pohlmann, 1996a). This test collection for Dutch text was used (a.o.) for the experiments with word conflation described in chapter 6. Section C.1 describes the document collection, C.2 documents the choices that were made to develop the test collection and the procedures that were used to ensure the quality of the test collection, finally section C.3 lists the set of queries.

C.1. UPLIFT Document set

The UPLIFT document collection consists of newspaper articles published in *Het Eindhovens Dagblad*, *Het Brabants Dagblad* and *Het Nieuwsblad* in the period January-October 1994. Some general statistics for the document collection are given in table C.1:

Total number of documents	59,608
Total number of words (tokens)	26,585,168
Total number of terms (types)	434,552
Size in megabytes	175
Max number of words per document	5,979
Av. number of words per document	446
Max number of terms per document	2,291
Av. number of terms per document	176

Table C.1. Statistics of the UPLIFT document collection

Figure C.1 gives a log-log view of the document length distribution, after stopping and stemming. We can see that the bulk of the documents have between 80-800 index terms. Unlike some of the TREC document collections, the UPLIFT collection does not contain extremely long documents.

C.2. UPLIFT topic creation and relevance assessments

C.2.1. Designing the evaluation experiment. The development of the UPLIFT collection was modeled after the protocols used for TREC. An important component of the TREC-style test collection development is the use of the pooling approach. Since a large collection of search engines, based on a variety of techniques was unavailable, there

[1] http://www-uilots.let.uu.nl/uplift/

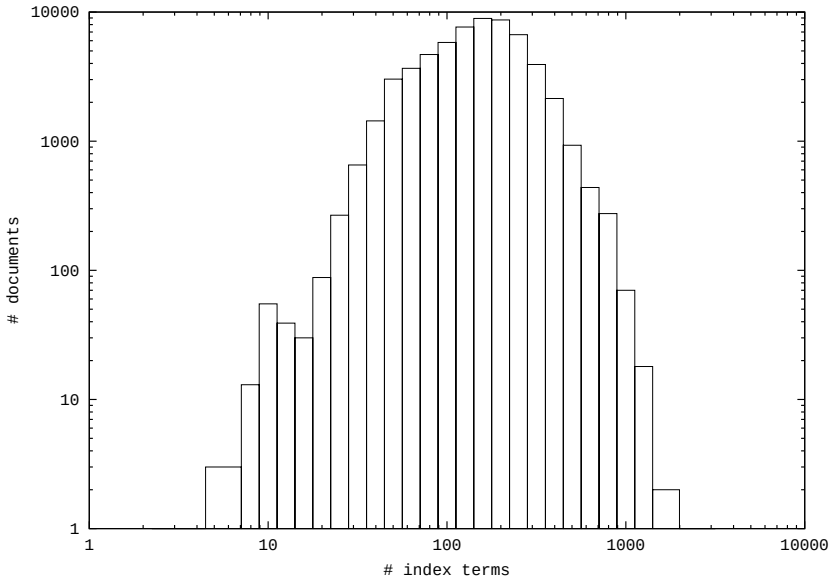

Figure C.1. Document length distribution of the UPLIFT corpus

were concerns about the quality of the pool of the UPLIFT test collection. Two pilot experiments were carried out in order to assess the scale of the problem.

We built several variants of our IR system (cf. Section 68), all based on query expansion. The expansion terms were selected through different forms of conflation, compound formation and synonymy. We had also built variants where we externally modified the query term weighting. The pilot experiment helped us to select a set of system variants which was as diverse as possible. We removed the weighting variants because their performance was not really different. We realized that only system versions that introduced new terms in the expansion would draw new relevant documents in the pool.

We also performed a pilot experiment with a search engine which was used to query the archive of a weekly mailing list (COLIBRI). This confirmed our intuition that (semi) automatic query expansion is a viable approach and that users tend to pose very short queries. Because we knew that the methods and retrieval models only show their full power with longer queries, we decided to require the users in the real experiment to formulate their search requests as full grammatical sentences. The pilot experiments are documented more extensively in (Kraaij & Pohlmann, 1996a)

C.2.2. Test users and test environment. Unlike the TREC evaluation procedure, the topic[2] creation process, the retrieval runs and the assessments were packaged together in a single session with a user (=assessor in TREC terminology). This method was motivated by the wish to have each topic created and evaluated by the same person. Secondly, in order to ensure a good pool quality, the user's query was run against a variety of systems in parallel automatically. The query was thus processed by 14 different system versions,

[2]In the UPLIFT case there is no distinction between a topic and a query.

resulting in 14 ranked lists of documents of length 1000 (=cutoff point). Test users did not see these separate lists, instead they worked with a list that consisted of a merge of the top 100 documents from each list, with duplicates removed. This resulted in a list ranging from 150 - 600 documents, depending on the query. This list was ordered on document number and presented to the user for relevance judgement. This merging and ordering method effectively hides the source of the document (i.e. the particular system version that retrieved it). In fact this setup was designed an on-line version of the TREC evaluation protocol, where the retrieval runs are done off-line at different sites.

The completely crossed design (each system is tested on all test queries) enabled a statistical analysis that separates run effects (the factor we are interested in) from query effects (cf. section 4.4). The test subjects for the experiment were recruited among staff and students of Utrecht University. Care was taken to ensure that subjects were not familiar with the details of the UPLIFT project (e.g., the specific hypotheses being tested in the experiment). Most subjects had no experience with web search engines or on-line IR systems. After some brief instruction (a short manual describing the task and some details about the document collection) subjects were asked to formulate a query in normal Dutch sentences. The choice to explicitly ask for full sentences / questions was motivated by several causes; we feared that most users would type very short queries (one - two words). This could result in a very heterogeneous query set (short and long). This was undesirable because it would add another source of variation. More importantly, we had the explicit goal to apply NLP techniques and assumed these would be more effective for full sentences. We hoped for example that full sentences would exhibit more morphological and syntactical variation than keyword queries. A less overt assumption was probably that " natural language" (and not a list of keywords) would be the preferred way to formulate a query for the average user, possibly via a speech interface. Finally, the requirement for a longer query would stimulate a user to formulate more precise questions, which could help to limit the number of relevant documents. Like TREC we took some precautions to avoid queries with zero or just a few relevant documents. Queries with just a few relevant documents have a very unstable average precision, obscuring the comparison of systems based on mean average precision. If a relevant document just differs a few ranks, this corresponds to huge differences in mean average precision for a topic with few relevant documents.

We initially collected 36 queries from 25 different test users. A second set of 30 queries from a new group of users was added later. In this thesis we will always refer to the full 66 topic test-collection, unless explicitly mentioned. This collection is listed in section C.3.

C.2.3. Description of a session. A test session was structured as follows:

(1) The test user was asked to read a manual which provides general background information about the experiment without giving away details about the different system versions. The manual described in detail what was expected from the test user:
(2) The user had to fill in name and run number, start and stop time are logged.
(3) The user had to enter a search query which must satisfy the following conditions:

- The query had to be stated in normal Dutch sentences.
- The query had to contain at least 15 words. (This is a heuristic to try to ensure that a sufficient number of content words is present in the query. This condition was not checked by the system though.)
- The query had to aim at a *set* (e.g., more than one) of relevant documents.

The manual contained a list of keywords related to topics in the database to give an idea of the scope of the database. Queries were not restricted to these topics.

(4) The system performed a test retrieval run and used a heuristic to test whether the user's query would yield enough relevant documents. The heuristic consisted of checking whether the score of the 50th ranked document was above a certain threshold[3]. If not, the user was asked to reformulate his query or to make up a new one.

(5) If the query had passed all tests, the n retrieval runs were executed and a merged list (cf. C.2.3) of potentially relevant documents was presented to the user. The tedious relevance judgement task was facilitated by a special application program which provided easy control and prevented errors.

After a post-hoc analysis of our data we found that the heuristic of step 4 was not reliable enough, since quite a few queries yielded just a few relevant documents. A better approach might have been to present the first 25 documents of the default system run, randomize them, let the user rate them and check whether there are enough relevant documents. If so, let the user assess the complete merged set with the already assessed documents left out.

C.3. UPLIFT QUERY COLLECTION

The UPLIFT collection contains 66 queries, which are reproduced without any post-editing:

(1) Geef mij alle artikelen die betrekking hebben op veeartsen, boeren, en ongevallen of misdaden in zowel Nederland als Belgie

(2) Welke verkiezingen vonden plaats? Welke partij heeft het goed gedaan? Welke partij heeft het slecht gedaan?

(3) Welke bosbranden hebben mensen het leven gekost?

(4) Tegen welke teams speelde het Nederlands elftal gelijk op de wereldkampioenschappen voetbal in de Verenigde Staten?

(5) Geef mij alle artikelen over een eventuele fusie van Berlicum met St. Michielsgestel.

(6) Geef alle artikelen met verslagen van rechtszaken met betrekking tot financiele compensatie voor medische fouten.

(7) Ik ben op zoek naar artikelen over de plannen voor de vorming van een stadsprovincie in de agglomeratie Eindhoven-Helmond

(8) Ik ben op zoek naar informatie over de procedure voor het verkrijgen van een verblijfsvergunning

[3]The engine applied query length normalisation, so scores were comparable across differences in query lengths.

(9) Voor welke literaire prijzen werden voornamelijk autobiografische romans genomineerd?
(10) Welke bekende personen werden naar aanleiding van het proces tegen Michael Jackson beschuldigd van seksuele perversiteiten met kinderen.
(11) Geef me de berichten die handelen over de plannen van het ministerie van onderwijs om de kosten van het promotieonderzoek door onderzoekers en assistenten in opleiding beheersbaar te maken en over de reacties van universiteiten
(12) Ik ben op zoek naar recensies van gespecialiseerde restaurants in de streek van Brabant Limburg en Utrecht met name vegetarische Indonesische en Italiaanse.
(13) Ik wilde graag wat meer weten over de voor- en nadelen van de verschillende methoden van afvalverwerking (in principe maakt het soort afval mij niet uit: papier chemisch afval en biologisch afval...), zoals daar zijn: verbranding compostering of dumping
(14) Wat voor voordelen heeft het gebruik van de electronische snelweg in universiteiten in Nederland opgeleverd?
(15) Op welke plaatsen langs welke snelwegen heeft de politie het afgelopen jaar omvangrijke snelheidscontroles uitgevoerd?
(16) Geef mij alle artikelen over peacekeeping operaties van de VN vredesmacht in Afrika en Azië.
(17) Hebben Tilburgse roeiers successen behaald ?
(18) In welke natuurgebieden in Nederland worden paarden en/of runderen gebruikt voor natuurlijk beheer.
(19) Geef mij informatie over de activiteiten waar RaRa zich in de afgelopen jaar mee bezig is geweest.
(20) Geef mij alle artikelen over de serie inbraken in cafe de Gouden Leeuw in Berlicum.
(21) In welke gemeenten zijn er plannen ontwikkeld voor een referendum over gemeentelijke herindeling en/of regiovorming?
(22) wat is de invloed van radioactieve straling op het lichamelijk en geestelijk functioneren van de mens?
(23) Wat was de mening van Groen Links over het wel of niet toestaan van euthanasie?
(24) Wat zijn de aanslagen die gepleegd werden in Israel het laaste jaar door Hamas en Il-Jihad Il-Islami
(25) geef een lijst van de aanvallen die Israel heeft gepleegd op zuid Libanon
(26) Geef mij een overzicht van de laatste ontwikkelingen op het gebied van de publieke en commerciële omroepen in Nederland
(27) Geef mij eens alle artikelen die over computers en talen en hun onderlinge verbanden gaan
(28) Het onderwerp moet zijn spoorwegmodelbouw of modelbouw in het algemeen, als hobby, met de nadruk op scenery en voertuigen. Is er een ruilbeurs of een manifestatie geweest?
(29) welke maatregelen worden in de biologische landbouw getroffen om energiebesparing te bewerkstelligen

(30) geef mij alle artikelen die er verschenen zijn van het satanisme in Noord-Brabant
(31) Welke stoffen in chemisch afval beinvloeden de vruchtbaarheid of bootsen oestrogenen na?
(32) denken de oostbrabantse veehouders het mestprobleem te kunnen oplossen door verbeterd veevoer of is volumebeleid noodzakelijk
(33) Welke landen heeft Beatrix een staatsbezoek gebracht?
(34) wat zijn de gevolgen van de overstromingen van de grote rivieren dit voorjaar geweest voor de landbouw en/of veeteelt in het getroffen gebied?
(35) in welke gemeente hebben de hindoestanen of andere allochtonen die partij zijn winst behaald bij de gemeenteraadsverkiezingen/?
(36) Geef mij alle verslagen van de wedstrijden van het Nederlands elftal op het WK voetbal in Amerika.
(37) Welke acties heeft Greenpeace in 1994 ondernomen?
(38) Ik ben op zoek naar gegevens over de werkgelegenheid in de gemeente Eindhoven.
(39) Ik vraag me af of het Nederlands Elftal ooit zo goed gespeeld heeft als op het WK Voetbal tegen Brazilie.
(40) Wat is het standpunt van de Europese Unie ten opzichte van een wereldwijd verbod op chemische wapens?
(41) Wanneer mag er radioactief materiaal uit een kerncentrale gehaald worden?
(42) Ik wilde de artikelen die betrekking hebben op vergoedingen van medische kosten (bv. door het ziekenfonds) voor het bevorderen van de zwangerschap: invitrofertilisatie (reageerbuisbevruchting) , kunstmatige inseminatie en andere technieken.
(43) Hebben jullie misschien artikelen over restaurants slijters en wijnimporteurs bv. ook over proeverijen in de streek?
(44) Is het alcoholgebruik bij allochtonen in de agglomeratie Eindhoven-Helmond de afgelopen jaren toe of juist af genomen door de opkomst van de house-muziek?
(45) Is de werkgelegenheid in de provincie Brabant de laatste jaren toegenomen?
(46) Ik wil informatie hebben over het beleid inzake industriele monumenten in Nederland.
(47) Wat zijn de gevolgen van het broeikaseffect voor de teelt van groente en fruit in Nederland?
(48) ik ben op zoek naar recensies over theater en toneel in Eindhoven.
(49) Ik ben op zoek naar artikelen over de gevolgen van de aanleg van de Betuwelijn voor het milieu.
(50) Ik wil graag meer informatie over de politieke jongerenorganisatie "CDJA".
(51) Wat is de situatie wat betreft de opvang van daklozen in Eindhoven Den Bosch en Tilburg?
(52) Ik wil graag weten welke asielzoekerscentra er in 1994 zijn ingericht en tot welke problemen ze hebben geleid.
(53) Hoe staat de samenleving tegenover condooms en wat voor invloed hebben condooms op de samenleving gehad ?
(54) Wat is er zoal over de behandeling van de mediawet in de tweede kamer geschreven?

(55) Worden in theatervoorstellingen meer onderwerpen door mannen dan door vrouwen aan de kaak gesteld ?
(56) Ik ben op zoek naar informatie over het gebruik van computers in de klassen en met name het gebruik daarvan in de basisscholen.
(57) Ik wil informatie over de toenemende criminaliteit onder jongeren in de Brabantse dorpen.
(58) Komen mensen die roken eerder in aanraking met drugs, dan niet-rokers?
(59) Wanneer en waarom is er al eens een verbod ingesteld op het vervoer van varkens in de regio Zuid-Brabant?
(60) Wat voor invloed heeft het tonen van geweld in films op de criminaliteit onder allochtone jongeren.
(61) Wat zijn de maatschappelijke gevolgen van aanslagen van extremistische organisaties.
(62) Ik zoek alle artikelen over politiek en guerilla in Latijns-Amerika en in het bijzonder Peru
(63) Ik ben op zoek naar artikelen over de uitbreiding van de Europese Unie.
(64) Ik zoek alle artikelen over het openbaar vervoer in Eindhoven
(65) Welke journalisten waren verdacht bij een aanslag van de RaRa?
(66) Ik wil graag alles weten over het ongeluk met de veerboot Estonia. Welke maatregelen zijn na dit ongeluk genomen om dergelijke rampen te voorkomen?

Bibliography

Aalbersberg, I. J., Brandsma, E., & Corthout, M. (1991). Full text document retrieval: from theory to applications. In Kempen, G., & de Vroomen, W., editors, *Informatiewetenschap 1991, Wetenschappelijke bijdragen aan de eerste STINFON-Conferentie*.

Allan, J., Ballesteros, L., Callan, J. P., Croft, W. B., & Lu, Z. (1996). Recent experiments with INQUERY. In Harman (1996). NIST Special Publication 500-236.

Allan, J., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E., Kraaij, W., Lafferty, J., Lavrenko, V., Lewis, D., Liddy, L., Manmatha, R., Aslam, J., McCallum, A., Ponte, J., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R., Belkin, N., Singhal, A., Smeaton, A., Turtle, H., Voorhees, E., Weischedel, R., Xu, J., Zhai, C., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., & Harman, D. (2003). Challenges in information retrieval and language modeling. *SIGIR Forum*, 37(1).

Allan, J., & Kumuran, G. (2003). Details on stemming in the language modeling framework. Technical Report IR-289, Center for Intelligent Information Retrieval, Amherst.

Apté, C., Damerau, F., & Weiss, S. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on information systems*, 12 No.3,233–251.

Arampatzis, A., & Hameren, A. (2001). The score-distributional threshold optimization for adaptive binary classification tasks. In Croft et al. (2001b), pp. 285–293.

Baayen, R. H., Piepenbrock, R., & van Rijn, H., editors (1993). *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia (PA).

Baeza-Yates, R., & Ribeiro-Neto, B., editors (1999). *Modern Information Retrieval*. Addison Wesley.

Ballesteros, L., & Sanderson, M. (2003). Addressing the lack of direct translation resources for cross-language retrieval. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*, pp. 147–152. ACM.

Ballesteros, L. A. (2000). Cross-language retrieval via transitive translation. In Croft, W. B., editor, *Advances in Information Retrieval*. Kluwer Academic Publishers.

Baumgarten, C. (1997). A probabilistic model for distributed information retrieval. In Belkin et al. (1997), pp. 258–266.

Baumgarten, C. (1999). A probabilistic solution to the selection and fusion problem in distributed information retrieval. In Hearst et al. (1999), pp. 246–253.

Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., & Järvelin, K., editors (2002). *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*. ACM Press.

Belkin, N. J., Ingwersen, P., & Leong, M., editors (2000). *Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*. ACM Press.

Belkin, N. J., Narasimhalu, A. D., & Willet, P., editors (1997). *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*. ACM Press.

Bell, T. C., Cleary, J. G., & Witten, I. H. (1990). *Text Compression*. Prentice Hall.

Berger, A., & Lafferty, J. (1999). Information retrieval as statistical translation. In Hearst et al. (1999), pp. 222–229.

Berger, A., & Lafferty, J. (2000). The Weaver system for document retrieval. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Blair, D. (1996). STAIRS redux: Thoughts on the STAIRS evaluation, ten years after. *Journal of the American Society for Information Science*, 47,4–22.

Blair, D., & Maron, M. (1985). An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM*, 20,289–299.

Bookstein, A., & Swanson, D. R. (1975). A decision theoretic foundation for indexing. *Journal of the American Society for Information Science*, 26,45–50.

Braschler, M., & Ripplinger, B. (2003). Stemming and decompounding for German text retrieval. In *Proceedings of ECIR2003, to appear*.

Braschler, M., Ripplinger, B., & Schäuble, P. (2002). Experiments with the Eurospider retrieval system for CLEF 2001. In Peters et al. (2002), pp. 45–50.

Braschler, M., & Schäuble, P. (2000). Using corpus-based approaches in a system for multilingual information retrieval. *Information Retrieval*, 3(273-284).

Braschler, M., & Schäuble, P. (2001). Experiments with the Eurospider retrieval system for CLEF 2000. In Peters (2001).

Broglio, J., Callan, J. P., Croft, W. B., & Nachbar, D. W. (1995). Document retrieval and routing using the INQUERY system. In Harman (1995), pp. 29–38. NIST Special Publication 500-236.

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2),79–85.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2),263–311.

Bryk, A., & Raudenbusch, S. (1992). *Hierarchical linear modelss*. Sage.

Buckley, C. (1999). personal communication.

Buckley, C., Mitra, M., Walz, J., & Cardie, C. (1998). Using clustering and superconcepts with SMART: TREC 6. In Voorhees, E. M., & Harman, D. K., editors, *The Sixth Text REtrieval Conference (TREC-6)*, volume 6. National Institute of Standards and Technology, NIST. NIST Special Publication 500-240.

Buckley, C., Mitra, M., Walz, J., & Cardie, C. (1999). SMART high precision: TREC 7. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC-3. In Harman (1995). NIST Special Publication 500-236.

Buckley, C., Singhal, A., Mitra, M., & Salton), G. (1996). New retrieval approaches using SMART: TREC4. In Harman (1996). NIST Special Publication 500-236.

Buckley, C., & Voorhees, E. (2000). Evaluating evaluation measure stability. In Belkin et al. (2000), pp. 33–40.

Burgin, R. (1992). Variations in relevance judgments and the evaluation of retrieval performance. *Information Processing & Management*, 28(5),619–627.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity based reranking for reordering documents and producing summaries. In Croft et al. (1998).

Cavnar, W. B. (1995). Using an n-gram-based document representation with a vector processing retrieval model. In Harman (1995), pp. 269–277. NIST Special Publication 500-236.

Cieri, C., Graff, D., Liberman, M., Martey, N., & Strassel, S. (2000). Large multilingual broadcast news corpora for cooperative research in topic detection and tracking: The TDT2 and TDT3 corpus efforts. *Proceedings of the Language Resources and Evaluation Conference (LREC2000)*.

Clarke, C. L., Cormack, G. V., & Tudhope, E. A. (1997). Relevance ranking for one to three term queries. In Devroye, L., & Chrisment, C., editors, *Proceedings of RIAO'97*, pp. 388–400.

Cleverdon, C. (1967). The cranfield tests on index language devices. In *Aslib Proceedings*, volume 19, pp. 173–192.

Conover (1980). *Practical Nonparametric Statistics*. John Wiley & Sons.

Cooper, W. (1971). The inadequacy of probability of usefulness as a ranking criterion for system retrieval output. Xeroxed, School of Library and Information Studies, University of California, Berkeley.

Cooper, W. S. (1994). The formalism of probability theory in IR: A foundation or an encumbrance. In Croft & van Rijsbergen (1994), pp. 242–247.

Crestani, F. (1998). *A study of the kinematics of probabilities in Information Retrieval*. PhD thesis, Department of Computing Science, University of Glasgow.

Crestani, F. (2000). personal communication.

Crestani, F., Lalmas, M., & van Rijsbergen, C. J., editors (1998a). *Information Retrieval: Uncertainty and Logics*. Kluwer Academic Publishers.

Crestani, F., Lalmas, M., van Rijsbergen, C. J., & Campbell, I. (1998b). "Iss this document relevant? . . . Probably": A survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4),528–552.

Crestani, F., Ruthven, I., Sanderson, M., & van Rijsbergen, C. (1996). The trouble with using a logical model of IR on a large collection of documents. In Harman (1996). NIST Special Publication 500-236.

Croft, B., Callan, J., & Lafferty, J. (2001a). Workshop on language modeling and information retrieval. *SIGIR FORUM*, 35(1).

Croft, W., Harper, D., D.H.Kraft, & Zobel, J., editors (2001b). *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*. ACM Press.

Croft, W., Moffat, A., van Rijsbergen, C., Wilkinson, R., & Zobel, J., editors (1998). *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*. ACM Press.

Croft, W., & van Rijsbergen, C., editors (1994). *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*. ACM Press.

Croft, W. B., & Xu, J. (1995). Corpus-specific stemming using word form co-occurence. In *Proceedings for the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR95)*, pp. 147–159.

Cronen-Townsend, S., Zhou, Y., & Croft, W. (2002). Predicting query performance. In Beaulieu et al. (2002).

Dagan, I., Church, K., & Gale, W. (1993). Robust bilingual word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives.*

Darwish, K., & Oard, D. W. (2003). Probabilistic structured query methods. In Callan, J., Cormack, G., Clarke, C., Hawking, D., & Smeaton, A., editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pp. 338–343. ACM Press.

de Heer, T. (1979). Quasi comprehension on natural language simulated by means of information traces. *Information Processing & Management*, 15,89–98.

de Vries, A. P. (2001). A poor man's approach to CLEF. In Peters (2001).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6),391–407.

Diekema, A. (2003). *Translation Events in cross-language information retrieval: lexical ambiguity, lexical holes, vocabulary mismatch, and correct translations*. PhD thesis, Syracuse University.

Doddington, G., & Fiscus, J. (2002). The 2002 topic detection and tracking (TDT2002) task definition and evaluation plan. Technical Report v. 1.1, National Institute of Standards and Technology.

Dumais, S. (1994). Panel: Evaluating interactive retrieval systems. In Croft & van Rijsbergen (1994).

Dumais, S. (1995). Using LSI for information filtering: TREC-3 experiments. In Harman (1995). NIST Special Publication 500-236.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, pp. 61–74.

Federico, M., & Bertoldi, N. (2002). Statistical cross-language information retrieval using n-best query translations. In Beaulieu et al. (2002), pp. 167–174.

Fiscus, J. G., & Doddington, G. R. (2002). Topic detection and tracking evaluation overview. In Allan, J., editor, *Topic Detection and Tracking*. Kluwer.

Foster, G. (2000). A maximum entropy / minimum divergence translation model. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Foster, G. F. (1991). Statistical Lexical Disambiguation. Msc. thesis, McGill University, School of Computer Science.

Fox, C. (1990). A stop list for general text. *SIGIR FORUM*, 24(1-2),19–35.

Fox, E., P.Ingwersen, & Fidel, R., editors (1995). *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*

(SIGIR '94). ACM Press.

Frakes, W. B., & Baeza-Yates, R., editors (1992). *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, NJ.

Franz, M., McCarley, J., & Roukos, S. (1999). Ad hoc and multilingual information retrieval at IBM. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Franz, M., McCarley, J. S., & Ward, R. T. (2000). Ad hoc, cross-language and spoken document retrieval at IBM. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Franz, M., McCarley, J. S., Ward, T., & Zhu, W.-J. (2001). Quantifying the utility of parallel corpora. In Croft et al. (2001b).

Frei, H.-P., Harman, D., Schäuble, P., & Wilkinson, R., editors (1996). *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)*. ACM Press.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal*, 35(3),233–245.

Fung, P. (2000). A statistical view on bilingual lexicon extraction, from parallel corpora to non-parallel corpora. In Véronis, J., editor, *Parallel Text Processing*, number 13 in Text Speech and Language Technology. Kluwer Academic Publishers.

Gale, W. A., Church, K. W., & Yarowsky, D. (1992). Work on statistical methods for word sense disambiguation. In *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*.

Goldsmith, J. (2001). Unsupervised learling of the morphology of a natural language. *Computational Linguistics*, 27(2),153–198.

Gollins, T., & Sanderson, M. (2001). Improving cross language retrieval with triangulated translation. In Croft et al. (2001b), pp. 90–95.

Gordon, M., & Pathak, P. (1999). Finding information on the world wide web: the retrieval effectiveness of search engines. *Information Processing & Management*, 35,141–180.

Grefenstette, G. (1998). The problem of cross-language information retrieval. In Grefenstette, G., editor, *Cross-Language Information Retrieval*, pp. 1–9. Kluwer Academic Publishers.

Greiff, W. R., & Morgan, W. T. (2003). Contributions of language modeling to the theory and practice of information retrieval. In Croft, W. B., & Lafferty, J., editors, *Language Modeling for Information Retrieval*, chapter 4. Kluwer Academic Publishers.

Grossman, D. A., & Frieder, O. (1998). *Information Retrieval, Algorithms and Heuristics*. Kluwer Academic Publishers.

Group, E. E. W. (1996). Evaluation of natural language processing systems. Technical report, ISSCO.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, 42(1),7–15.

Harman, D. (1992). Relevance feedback and other query modification techniques. In Frakes, W. B., & Baeza-Yates, R., editors, *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, NJ.

Harman, D. K., editor (1995). *The Third Text REtrieval Conference (TREC-3)*, volume 4. National Institute of Standards and Technology, NIST. NIST Special Publication 500-236.

Harman, D. K., editor (1996). *The Fourth Text REtrieval Conference (TREC-4)*, volume 4. National Institute of Standards and Technology, NIST. NIST Special Publication 500-236.

Harter, S. (1975). A probabilistic approach to automatic keyword indexing. *Journal of the American Society for Information Science*, 26,197–206 and 280–289.

Hayes, W. L. (1981). *Statistics*. Holt, Rinehart and Winston.

Hearst, M., Gey, F., & Tong, R., editors (1999). *Proceedings of the 22th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM Press.

Hersh, W., Turpin, A., Price, S., Chan, B., Kraemer, D., Sacherek, L., & Olson, D. (2000). Do batch and user evaluations give the same results? In Belkin et al. (2000), pp. 17–24.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In Nicolaou, C., & Stephanides, C., editors, *Research and Advanced Technology for Digital Libraries - Second European Conference, ECDL'98, Proceedings*, number 1513 in Lecture Notes in Computer Science. Springer Verlag.

Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. PhD thesis, University of Twente.

Hiemstra, D., & de Jong, F. (1999). Disambiguation strategies for cross-language information retrieval. In *European Conference on Digital Libraries*, pp. 274–293.

Hiemstra, D., de Jong, F., & Kraaij, W. (1997). A domain specific lexicon acquisition tool for cross-language information retrieval. In Devroye, L., & Chrisment, C., editors, *Proceedings of RIAO'97*, pp. 217–232.

Hiemstra, D., & Kraaij, W. (1999). Twenty-one at TREC-7: Ad hoc and cross language track. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Hiemstra, D., & Kraaij, W. (2005). A language modeling approach for TREC. In Voorhees, E. M., & Harman, D., editors, *TREC: Experiment and Evaluation in Information Retrieval*. MIT press. forthcoming.

Hiemstra, D., Kraaij, W., Pohlmann, R., & Westerveld, T. (2001a). Translation resources, merging strategies and relevance feedback. In Peters, C., editor, *Cross-Language Information Retrieval and Evaluation*, number 2069 in Lecture Notes in Computer Science. Springer Verlag.

Hiemstra, D., Kraaij, W., Pohlmann, R., & Westerveld, T. (2001b). Twenty-One at CLEF-2000: Translation resources, merging strategies and relevance feedback. In Peters (2001).

Hofmann, T. (1999). Probabilistic latent semantic indexing. In Hearst et al. (1999), pp. 50–57.

Hollink, V., Kamps, J., Monz, C., & de Rijke, M. (2003). Monolingual document retrieval for european languages. *Information Retrieval*.

Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In Korfhage et al. (1993), pp. 329–338.

Hull, D. (1996). Stemming algorithms – a case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1).

Hull, D. (1997). Using structured queries for disambiguation in cross-language information retrieval. In Hull, D., & Oard, D., editors, *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence.

http://www.clis.umd.edu/dlrg/filter/sss/papers/.

Hull, D., Kantor, P. B., & Ng, K. B. (1999). Advanced approaches to the statistical analysis of TREC information retrieval experiments. Unpublished Report.

Hull, D. A., & Grefenstette, G. (1996). Querying across languages: A dictionary-based approach to multilingual information retrieval. In Frei et al. (1996), pp. 49–57.

Jacquemin, C., & Tzoukermann, E. (1999). NLP for term variant extraction: Synergy between morphology, lexicon and syntax. In Strzalkowski, T., editor, *Natural Language Information Retrieval*. Kluwer Academic Publishers.

Järvelin, K., & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In Belkin et al. (2000), pp. 41–48.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press.

Jin, H., Schwartz, R., Sista, S., & Walls, F. (1999). Topic tracking for radio, tv broadcast and newswire. In *Proceedings of the DARPA Broadcast News Workshop*.

Jing, H., & Tzoukermann, E. (1999). Information retrieval based on context distance and morphology. In Hearst et al. (1999), pp. 90–96.

Jing, Y., & Croft, W. B. (1994). An association thesaurus for information retrieval. In *Proceedings of RIAO-94, 4th International Conference "Recherche d'Information Assistee par Ordinateur"*, pp. 146–160, New York, US.

Jones, K. S., & Robertson, S. (2001). LM vs PM: Where's the relevance. In Callan, J., Croft, B., & Lafferty, J., editors, *Proceedings of the workshop on Language Modeling and Information Retrieval*.

Jourlin, P. (2000). personal communication.

Jourlin, P., Johnson, S. E., Sparck Jones, K., & Woodland, P. C. (1999). General query expansion techniques for spoken document retrieval. In *Proceedings of the ESCA ETRW workshop: Accessing Information in Spoken Audio*.

Joyce, T., & Needham, R. (1958). The thesaurus approach to information retrieval. *American Documentation*, 9,192–197.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing*. Prentice-Hall.

Kando, N., & Nozue, T., editors (1999). *NTCIR Workshop 1: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition. Tokyo, Japan*. http: //www.rd.nacsis.ac.jp/ ntcadm/workshop/OnlineProceedings/.

Kantor, P. B., & Voorhees, E. (1997). Report on the TREC-5 confusion track. In Voorhees & Harman (1997), pp. 65–74. NIST Special Publication 500-238.

Keenan, S., Smeaton, A., & Keogh, G. (2001). The effect of pool depth on system evaluation in TREC. *Information Processing & Management*, 52(7),570–573.

Kekäläinen, J., & Järvelin, K. (2000). The co-effects of query structure and expansion on retrieval performance in probabilistic text retrieval. *Information Retrieval*, 1(4),329–342.

Kilgariff, A., & Grefenstette, G. (2003). Introduction to the special issue on the web as a corpus. *Computational Linguistics*, 29(3).

Knight, K., & Graehl, J. (1997). Machine transliteration. In Cohen, P. R., & Wahlster, W., editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 128–135, Somerset, New Jersey. Association for Computational Linguistics.

Korfhage, R., Rasmussen, E., & Willett, P., editors (1993). *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*. ACM Press.

Kraaij, W. (2002). TNO at CLEF-2001: Comparing translation resources. In Peters et al. (2002).

Kraaij, W. (2003). Exploring transitive translation methods. In Vries, A. P. D., editor, *Proceedings of DIR 2003*.

Kraaij, W., & de Jong, F. (2004). Transitive probabilistic clir models. In *Proceedings of RIAO 2004*.

Kraaij, W., Nie, J.-Y., & Simard, M. (2003). Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3),381–419.

Kraaij, W., & Pohlmann, R. (1994). Porter's stemming algorithm for Dutch. In Noordman, L., & de Vroomen, W., editors, *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pp. 167–180.

Kraaij, W., & Pohlmann, R. (1995). Evaluation of a Dutch stemming algorithm. In Rowley, J., editor, *The New Review of Document & Text Management*, volume 1, pp. 25–43. Taylor Graham Publishing, London.

Kraaij, W., & Pohlmann, R. (1996a). Using linguistic knowledge in information retrieval. OTS Working Paper OTS-WP-CL-96-001, Research Institute for Language and Speech (OTS), Utrecht University.

Kraaij, W., & Pohlmann, R. (1996b). Viewing stemming as recall enhancement. In Frei et al. (1996), pp. 40–48.

Kraaij, W., & Pohlmann, R. (1998). Comparing the effect of syntactic vs. statistical phrase indexing strategies for Dutch. In Nicolaou, C., & Stephanides, C., editors, *Research and Advanced Technology for Digital Libraries - Second European Conference, ECDL'98, Proceedings*, number 1513 in Lecture Notes in Computer Science, pp. 605–614. Springer Verlag.

Kraaij, W., & Pohlmann, R. (2001). Different approaches to cross language information retrieval. In Daelemans, W., Sima'an, K., Veenstra, J., & Zavrel, J., editors, *Computational Linguistics in the Netherlands 2000*, number 37 in Language and Computers: Studies in Practical Linguistics, pp. 97–111, Amsterdam. Rodopi.

Kraaij, W., Pohlmann, R., & Hiemstra, D. (2000). Twenty-one at TREC-8: using language technology for information retrieval. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Kraaij, W., & Spitters, M. (2003). Language models for topic tracking. In Croft, B., & Lafferty, J., editors, *Language Models for Information Retrieval*. Kluwer Academic Publishers.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In Beaulieu et al. (2002).

Krovetz, R. (1993). Viewing morphology as an inference process. In Korfhage et al. (1993), pp. 191–203.

Krovetz, R., & Croft, W. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2),115–141. opz.

Kwok, K. (2000). TREC-8 ad-hoc, query and filtering track experiments using PIRCS. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Lafferty, J., & Zhai, C. (2001a). Document language models, query models, and risk minimization for information retrieval. In Croft et al. (2001b).

Lafferty, J., & Zhai, C. (2001b). Probabilistic IR models based on document and query generation. In Callan, J., Croft, B., & Lafferty, J., editors, *Proceedings of the workshop on Language Modeling and Information Retrieval*.

Laffling, J. (1992). On constructing a transfer dictionary for man and machine. *Target*, 4(4),17–31.

Lancaster, W. (1969). MEDLARS: Report on the evaluation of its operating efficiency. *American Documentation*, 20,119–142.

Lavrenko, V., Allen, J., DeGuzman, E., LaFlamme, D., Pollard, V., & Thomas, S. (2002a). Relevance models for topic detection and tracking. In *Proceedings of HLT 2002*.

Lavrenko, V., Choquette, M., & Croft, W. (2002b). Cross-lingual relevance models. In Beaulieu et al. (2002).

Lavrenko, V., & Croft, B. (2003). Relevance models in information retrieval. In Croft, B., & Lafferty, J., editors, *Language Models for Information Retrieval*, pp. 11–56. Kluwer Academic Publishers.

Lavrenko, V., & Croft, W. (2001). Relevance-based language models. In Croft et al. (2001b).

Leek, T., Jin, H., Sista, S., & Schwartz, R. (1999). The BBN crosslingual topic detection and tracking system. In Fiscus, J., editor, *Proceedings of the TDT-3 workshop*.

Leek, T., Schwartz, R., & Sista, S. (2002). Probabilistic approaches to topic detection and tracking. In Allan, J., editor, *Topic Detection and Tracking*. Kluwer.

Lehtokangas, R., & Airio, E. (2002). Translation via a pivot language challenges direct translation in CLIR. In *Proceedings of the SIGIR 2002 Workshop: Cross-Language Information Retrieval: A Research Roadmap*.

Leighton, H. V., & Srivastava, J. (1999). First 20 precision among world wide web search services (search engines). *Journal of the American Society for Information Science*, 50(10),870–881.

Lesk, M., Harman, D., Fox, E., Wu, H., & Buckey, C. (1997). The SMART lab report. *SIGIR FORUM*, 31(1).

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals,. *Sov. Phys, Dokl.*, 10,707–710.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in informatiion retrieval. In C.Nédellec, & Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 4–15.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11,22–31.

Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM journal*, pp. 309–317.

Manmatha, R., Rath, T., & Feng, F. (2001). Modelling score distributions for combining the outputs of search engines. In Croft et al. (2001b), pp. 267–275.

Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT press.

Maron, M., & Kuhns, J. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery*, 7,216–244.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying news stories using memory-based reasoning. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92)*, pp. 59–65. ACM Press.

Maxwell, S. E., & Delaney, H. D. (1990). *Designing Experiments and Analyzing Data*. Wadsworth Publishing Company.

Mayfield, J., & McNamee, P. (1999). Indexing using both n-grams and words. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

McCarley, J. S., & Roukos, S. (1998). Fast document translationa for cross-language information retrieval. In Farwell, D., Gerber, L., & Hovy, E., editors, *Machine Translation and the Information Soup, Third Conference of the Association for Machine Translation in the Americas, AMTA'98*, number 1529 in Lecture Notes in Artificial Intelligence. Springer.

McNamee, P., & Mayfield, J. (2001). A language-independent approach to european text retrieval. In Peters, C., editor, *Cross-Language Information Retrieval and Evaluation*, number 2069 in Lecture Notes in Computer Science. Springer Verlag.

McNamee, P., & Mayfield, J. (2002). Comparing cross-lanuage query expansion techniques by degrading translation reources. In Beaulieu et al. (2002).

Meghini, C., Sebastiani, F., & Straccia, U. (1998). Mirlog: a logic for multimedia information retrieval. In Crestani, F., Lalmas, M., & van Rijsbergen, C. J., editors, *Information Retrieval: Uncertainty and Logics*. Kluwer Academic Publishers.

Mihalcea, R., & Moldovan, D. (2000). Semantic indexing using wordnet senses. In *Proceedings of ACL Workshop on IR & NLP*.

Miller, D. R., Leek, T., & Schwartz, R. M. (1999a). BBN at TREC-7: Using hidden markov models for information retrieval. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Miller, D. R. H., Leek, T., & Schwartz, R. M. (1999b). A hidden markov model information retrieval system. In Hearst et al. (1999), pp. 214–221.

Miller, G., Newman, E., & Friedman, E. (1957). Some effect of intermittent silence. *American Journal of Psychology*, 70,311–313.

Miller, G. A. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4),235–312.

Mitchell, T. (1996). *Machine Learning*. McGraw-Hill.

Mitra, M., Buckley, C., Singhal, A., & Cardie, C. (1997). An analysis of statistical and syntactic phrases. In Devroye, L., & Chrisment, C., editors, *Proceedings of RIAO'97*, pp. 200–214.

Mittendorf, E. (1998). *Data Corruption and Information Retrieval*. PhD thesis, ETH Zürich.

Mittendorf, E., & Schäuble, P. (1994). Document and passage retrieval based on hidden markov models. In Croft & van Rijsbergen (1994), pp. 318–327.

Monz, C. (2003). *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam.

Mooers, C. (1952). Information retrieval viewed as temporal signalling. In *Proceedings of the International Conference of Mathematicians, Cambridge Massachusettes 1950*, pp. 572–573.

Ng, H. T., Goh, W. B., & Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In Belkin et al. (1997), pp. 67–73.

Ng, K. (2000a). A maximum likelihood ratio information retrieval model. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Ng, K. (2000b). *Subword-based Approaches for Spoken Document Retrieval*. PhD thesis, M.I.T.

Nie, J., Simard, M., Isabelle, P., & Durand, R. (1999). Cross-language information retrieval based on parallel texts and automatic mining of parallel texts in the web. In Hearst et al. (1999), pp. 74–81.

Nie, J.-Y., & Lepage, F. (1998). Toward a broader logical model for information retrieval. In Crestani, F., Lalmas, M., & van Rijsbergen, C. J., editors, *Information Retrieval: Uncertainty and Logics*. Kluwer Academic Publishers.

Oard, D. (1998). A comparative study of query and document translation for cross-language information retrieval. In *Proceedings of AMTA 1998*, pp. 472–483.

Oard, D. W. (1997). Alternative approaches for cross-language text retrieval. In Hull, D., & Oard, D., editors, *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence. http://www.clis.umd.edu/dlrg/filter/sss/papers/.

Oard, D. W., & Dorr, B. J. (1996). Survey of multilingual text retrieval. Technical Report UMIACS-TR-96-19, University of Maryland, Institute for Advanced Computer Studies. http://www.ee.umd.edu/medlab/filter/papers/mlir.ps.

Oard, D. W., & Gonzalo, J. (2002). The CLEF 2001 interactive track. In Peters et al. (2002), pp. 203–214.

Ogilvie, P., & Callan, J. (2001). Experiments using the LEMUR toolkit. In Voorhees, E. M., & Harman, D. K., editors, *The Tenth Text REtrieval Conference (TREC-2001), notebook*, volume 10. National Institute of Standards and Technology, NIST.

Over, P. (1997). TREC-5 interactive track report. In Voorhees & Harman (1997). NIST Special Publication 500-238.

Perry, J. W., Kent, A., & Berry, M. (1956). *Machine Literature Searching*. InterScience, NY.

Peters, C., editor (2001). *Cross-Language Information Retrieval and Evaluation*, number 2069 in Lecture Notes in Computer Science. Springer Verlag.

Peters, C., Braschler, M., Gonzalo, J., & Kluck, M., editors (2002). *Evaluation of Cross-Language Information Retrieval Systems, Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*. Springer.

Pirkola, A. (1998). The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In Croft et al. (1998), pp. 55–63.

Pirkola, A., & Järvelin, K. (2001). Employing the resolution power of search keys. *Information Processing & Management*, 52(7),575–583.

Pirkola, A., Keskustalo, H., & Järvelin, K. (1999). The effects of conjunction, facet structure and dictionary combinations in concept-based cross-language retrieval. *Information Retrieval*, 1(3),217–250.

Pohlmann, R., & Kraaij, W. (1997a). The effect of syntactic phrase indexing on retrieval performance for Dutch texts. In Devroye, L., & Chrisment, C., editors, *Proceedings of RIAO'97*, pp. 176–187.

Pohlmann, R., & Kraaij, W. (1997b). Improving the precision of a text retrieval system with compound analysis. In Landsbergen, J., Odijk, J., van Deemter, K., & van Zanten, G. V., editors, *CLIN VII – Papers from the Seventh CLIN meeting*, pp. 115–128.

Ponte, J. M. (2001). Is information retrieval anything more than smoothing. In *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR2001)*.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In Croft et al. (1998), pp. 275–281.

Popovič, M., & Willett, P. (1992). The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5),384–390.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3),130–137.

Qiu, Y. (1995). *Automatic Query expansion Based on a Similarity Thesaurus*. PhD thesis, ETH Zürich.

Raghavan, V. V., & Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7(3),205–229.

Raghavan, V. V., & Wong, S. K. M. (1986). A critical analysis of the vector space model for information retrieval. *Journal of the American Society for Information Science*, 37,279–287.

Resnik, P. (1998). Parallel stands: A preliminary investigation into mining the web for bilingual text. In *Proceedings of AMTA*, number 1529 in Lecture Notes in Artificial Intelligence, pp. 72–82.

Rijsbergen, C. J. v. (1979). *Information Retrieval*. Butterworths, London.

Riloff, E. (1995). Little words can make a big difference for text classification. In Fox et al. (1995), pp. 130–136.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33,294–304.

Robertson, S., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3),129–146.

Robertson, S., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In Croft & van Rijsbergen (1994), pp. 232–241.

Robertson, S., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1).

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System*. Prentice Hall.

Rölleke, T., & Fuhr, N. (1998). Information retrieval with probabilistic datalog. In Crestani, F., Lalmas, M., & van Rijsbergen, C. J., editors, *Information Retrieval: Uncertainty and Logics*. Kluwer Academic Publishers.

Ruiz, M., Diekema, A., & Sheridan, P. (2000). CINDOR conceptual interlingua document retrieval. In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Salton, G. (1968). *Automatic Information Organisation and Retrieval*. McGraw-Hill, New York. opz.

Salton, G. (1973). Experiments in multi-lingual information retrieval. *Information Processing & Management*, 2(1).

Salton, G. (1989). *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, Reading (MA).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5),513–523.

Salton, G., Fox, E., & Wu, H. (1983). Extended boolen information retrieval. *Communications of the ACM*, 26(12),1022–36.

Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., New York.

Salton, G., Wong, A., & Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18,613–620.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In Croft & van Rijsbergen (1994), pp. 142–152.

Sanderson, M. (2000). Retrieving with good sense. *Information Retrieval*, 2(1).

Sanderson, M., & van Rijsbergen, C. (1999). The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems*, 17(4),440–465.

Saracevic, T. (1975). Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26(6),321–343.

Savoy, J. (1997). Statistical inference in retrieval effectiveness evaluation. *Information Processing & Management*, 33(4),495–512.

Savoy, J. (2002). Report on CLEF-2001 experiments. In Peters et al. (2002).

Schäuble, P. (1989). *Information Retrieval Based on Information Structures*. PhD thesis, ETH Zürich.

Schütze, H., Hull, D., & Pedersen, J. (1995). A comparison of classifiers and document representations for the routing problem. In Fox et al. (1995), pp. 229–237.

Schütze, H., & Pedersen, J. O. (1995). Information retrieval based on word senses. In *Proceedings for the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR95)*, pp. 161–175.

Selberg, E., & Etzioni, O. (2000). On the instability of web search engines. In Mariani, J., & Harman, D., editors, *Proceedings of RIAO'2000*, pp. 223–233.

Shannon, C. E., & Weaver, W. (1949). *The Mathematical Theory of Communication*. The University of Illinois Press, Urbana.

Sheridan, P., & Ballerini, J. (1996). Experiments in multi-lingual information retrieval using the SPIDER system. In Frei et al. (1996), pp. 58–65.

Simard, M., Foster, G., & Isabelle, P. (1992). Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodogical Issues in Machine translation (TMI92)*.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted document length normalization. In Frei et al. (1996), pp. 21–29.

Singhal, A., Choi, J., Hindle, D., Lewis, D., & Perreira, F. (1999). AT&T at TREC-7. In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Singhal, A., & Pereira, F. (1999). Document expansion for speech retrieval. In Hearst et al. (1999), pp. 34–41.

Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1995). Document length normalization. Technical Report TR95-1529, Cornell University.

Snedecor, G. W., & Cochran, W. G. (1980). *Statistical Methods*. Iowa State University Press.

Sormunen, E. (2000). A novel method for the evaluation of boolean query effectiveness across a wide operational range. In Belkin et al. (2000), pp. 25–32.

Sparck Jones, K. (1974). Automatic indexing. *Journal of Documentation*.

Sparck Jones, K., editor (1981). *Information Retrieval Experiment*. Butterworths, London.

Sparck Jones, K. (1992). Assumptions and issues in text-based retrieval. In Jacobs, P. S., editor, *Text-Based Intelligent Systems - Current Research and Practice in Information Exctraction and Retrieval*. Lawrence Erlbaum Assc., Hillsdale, New Jersey.

Sparck Jones, K. (1999). What is the role of NLP in text retrieval. *In: T.Strzalkowski (ed.), Natural Language Information Retrieval*, pp. 1–22.

Sparck Jones, K., Walker, S., & Robertson, S. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing & Management*, 36(6).

Sparck Jones, K., & Willett, P. (1997a). Key concepts. In Sparck Jones, K., & Willett, P., editors, *Readings in Information Retrieval*, chapter 1, pp. 85–91. Morgan Kaufmann Publishers.

Sparck Jones, K., & Willett, P. (1997b). Overall introduction. In Sparck Jones, K., & Willett, P., editors, *Readings in Information Retrieval*, chapter 1, pp. 1–7. Morgan Kaufmann Publishers.

Sparck Jones, K., & Willett, P., editors (1997c). *Readings in Information Retrieval*. Morgan Kaufmann Publishers.

Spitters, M., & Kraaij, W. (2001). Using language models for tracking events of interest over time. In *Proceedings of the Workshop on Language Models for Information Retrieval (LMIR2001)*.

Spitters, M., & Kraaij, W. (2002). Unsupervised event clustering in multilingual news streams. *Proceedings of the LREC2002 Workshop on Event Modeling for Multilingual Document Linking*, pp. 42–46.

Strzalkowski, T. (1995). Natural language information retrieval. *Information Processing & Management*, 31(3),397–417.

Strzalkowski, T., Lin, F., Wang, J., Guthrie, L., Leistensnider, J., Wilding, J., Karlgren, J., Straszheim, T., & Perez-Carballo, J. (1997). Natural language information retrieval: TREC-5 report. In Voorhees & Harman (1997). NIST Special Publication 500-238.

Stuart, A., & Ord, J. K. (1987). *Kendall's Advanced Theory of Statistics*. Charles Griffin & Company Limited.

Swets, J. (1969). Effectiveness of information retrieval methods. *American Documentation*, 20(1),72–89.

Tague, J. M. (1981). The pragmatics of information retrieval experimentation. In Sparck Jones, K., editor, *Information Retrieval Experiment*, pp. 59–102. Butterworths.

Tague-Sutcliffe, J. (1995). *Measuring Information, An Information Services Perspective*. Academic Press, San Diego (CA).

Tague-Sutcliffe, J., & Blustein, J. (1995). A statistical analysis of the TREC-3 data. In Harman (1995), pp. 385–398. NIST Special Publication 500-236.

ter Stal, W. (1996). *Automated Interpretation of Nominal Compounds in a Technical Domain*. PhD thesis, Technische Universiteit Twente, UT Repro, Enschede.

ter Stal, W., Beijert, J.-H., de Bruin, G., van Gent, J., de Jong, F., Kraaij, W., Netter, K., & Smart, G. (1998). Twenty-one: cross-language disclosure and retrieval of multimedia documents on sustainable development. *Computer Networks and ISDN Systems*, 30(13),1237–1248.

Tukey, J. W. (1953). The problem of multiple comparisons. Unpublished manuscript reprinted in H.I. Braun (ed.) The collected works of John W. Tukey: Vol VIII Multiple comparisons: 1948-193, (1994), pp 1-300. Chapman and Hall, New York, NY.

Turtle, H. R. (1991). *Inference Networks for Document Retrieval*. PhD thesis, CIIR, University of Massachusettes.

van Rijsbergen, C. (1986). A non-classical logic for information retrieval. *Computer Journal*, 29,481–485.

Véronis, J., editor (2000). *Parallel Text Processing*. Kluwer Academic Publishers.

Voorhees, E., & Harman, D. (1999a). Overview of the seventh text retrieval conference (trec-7). In Voorhees & Harman (1999b). NIST Special Publication 500-242.

Voorhees, E., & Harman, D. (2000a). Overview of the eighth Text REtrieval Conference (TREC-8). In Voorhees & Harman (2000b). NIST Special Publication 500-246.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In Croft & van Rijsbergen (1994), pp. 61–69.

Voorhees, E. M. (1998). Variations in relevance judgements and the measurement of retrieval effectiveness. In Croft et al. (1998), pp. 315–323.

Voorhees, E. M., & Harman, D. K., editors (1997). *The Fifth Text REtrieval Conference (TREC-5)*, volume 5. National Institute of Standards and Technology, NIST. NIST Special Publication 500-238.

Voorhees, E. M., & Harman, D. K., editors (1999b). *The Seventh Text REtrieval Conference (TREC-7)*, volume 7. National Institute of Standards and Technology, NIST. NIST Special Publication 500-242.

Voorhees, E. M., & Harman, D. K., editors (2000b). *The Eigth Text REtrieval Conference (TREC-8)*, volume 8. National Institute of Standards and Technology, NIST. NIST Special Publication 500-246.

Vosse, T. G. (1994). *The Word Connection*. PhD thesis, Rijksuniversiteit Leiden, Neslia Paniculata Uitgeverij, Enschede.

Wayne, C. (2000). Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. *Proceedings of the Language Resources and Evaluation Conference (LREC2000)*, pp. 1487–1494.

Womser-Hacker, C. (2002). Multilingual topic generation within the CLEF 2001 experiments. In Peters et al. (2002), pp. 389–393.

Wong, S. K. M., & Yao, Y. Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1),38–68.

Wong, S. K. M., Ziarko, W., Raghavan, V. V., & Wong, P. C. N. (1986). On extending the vector space model for boolean query processing. In *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '86)*, pp. 175–185. ACM Press.

Wong, S. K. M., Ziarko, W., Raghavan, V. V., & Wong, P. C. N. (1987). On modeling of information retrieval concepts in vector space. *TODS*, 12(2),299–321.

Xu, J., Fraser, A., & Weischedel, R. (2002a). Empirical studies in strategies for Arabic retrieval. In Beaulieu et al. (2002).

Xu, J., Fraser, A., & Weischedel, R. (2002b). TREC 2001 cross-lingual retrieval at BBN. In Voorhees, E. M., & Harman, D. K., editors, *The Tenth Text REtrieval Conference (TREC-10)*, volume 10. National Institute of Standards and Technology, NIST.

Xu, J., Weischedel, R., & Nguyen, C. (2001). Evaluating a probabilistic model for cross-lingual information retrieval. In Croft et al. (2001b).

Yang, J., & Lange, E. D. (1998). SYSTRAN on Alta Vista: A user study on real-time machine translation on the internet. In Farwell, D., Gerber, L., & Hovy, E., editors, *Machine Translation and the Information Soup, Third Conference of the Association for Machine Translation in the Americas, AMTA'98*, number 1529 in Lecture Notes in Artificial Intelligence. Springer.

Yang, Y., Brown, R. D., Frederking, R. E., Carbonell, J. G., Geng, Y., & Lee, D. (1997). Bilingual corpus based approaches to translingual information retrieval. In *Proceedings of the The 2nd Workshop on "Multilinguality in Software Industry: The AI Contribution (MULSAIC'97)"*.

Yang, Y., Carbonell, J. G., , Brown, R. D., & Frederking, R. E. (1998). Translingual information retrieval: Learning from bilingual corpora. *Artificial Intelligence*, 103(1–2),323–345.

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of COLING 2000*, pp. 947–953.

Yu, C., Buckley, C., Lam, K., & Salton, G. (1983). A generalized term dependence model in information retrieval. *Information Processing & Management*, 2,129–154.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8,338–353.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In Croft et al. (2001b).

Zhai, C., Tong, X., Milić-Frayling, N., & Evans, D. (1997). Evaluation of syntactic phrase indexing – CLARIT NLP track report. In Voorhees & Harman (1997). NIST Special Publication 500-238.

Zipf, G. K. (1949). *Human behaviour and the principle of the least effort*. Addison Wesley.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments. In Croft et al. (1998), pp. 307–314.

Zobel, J., & Moffat, A. (1998). Exploring the similarity space. *SIGIR FORUM*, 32(1),18–34.

Index

S

T

Summary

Search engine technology builds on theoretical and empirical research results in the area of information retrieval (IR). This dissertation makes a contribution to the field of language modeling (LM) for IR, which views both queries and documents as instances of a unigram language model and defines the matching function between a query and each document as the probability that the query terms are generated by the document language model. The work described is concerned with three research issues.

The first research question addressed is how linguistic resources can be optimally combined with statistical language models. A case study on embedding morphological normalization for Dutch shows that complex models for matching in word form space are less effective than a simple model based on matching in the reduced feature space of word stems. A case study on cross-language information retrieval (CLIR) shows that probabilistic retrieval models with fully integrated statistical translation perform significantly better than the frequently applied synonym-based approach. A crucial element is the fact that the probabilistic models can accommodate multiple weighted translation variants, which is especially effective when translations are derived from parallel corpora.

The second research issue is an investigation of the hypothesis that it should be possible to formulate a single LM-based document ranking formula, which is effective for both topic tracking and ad hoc search. The first task differs from the latter by the fact that ranking score distributions for different topics must be comparable on an absolute scale. A variant model which meets this criterion is proposed and its relationship to the classical odds-of-relevance model and the Kullback-Leibler divergence is explained. The model is based on the reduction in cross-entropy associated with a certain document model in comparison to a background model. Besides being an adequate and unifying model for topic tracking and ad hoc search, the cross-entropy based approach also allows for intuitive modeling of the CLIR task by mapping either the query or document language model onto a language model in a different language.

The final research issue concerns a more general problem for IR researchers, namely statistical validation of experimental results. Standard statistical significance tests are reviewed and their underlying assumptions about the properties of test data are validated on actual data from IR experiments. A set of guidelines is presented, which contribute to an improved methodological framework. In addition, a fairly comprehensive discussion of state-of-the art IR models and techniques is included, which can be read as a tutorial text.

Samenvatting

Alledaagse zoektechnologie is gebaseerd op theoretisch en praktisch onderzoek uit het vakgebied "Information Retrieval" (IR). Dit proefschrift draagt bij aan de ontwikkeling van het deelgebied statistische taalmodellering voor IR. Bij een IR-systeem dat gebaseerd is op een dergelijke aanpak wordt een document gerepresenteerd als een ongeordende verzameling van de woorden waaruit het document is opgebouwd. Vervolgens wordt die verzameling getransformeerd tot een kansmodel, door voor ieder uniek woord het aantal keren dat het woord voorkomt in het document te delen door het totaal aantal woorden. Voor een bepaalde zoekvraag kunnen documenten vervolgens in volgorde van relevantie worden geplaatst door voor ieder document te bepalen hoe groot de kans is dat de zoekvraag door het kansmodel van het document wordt gegenereerd. Dit proefschrift concentreert zich op drie onderwerpen.

Het eerste onderwerp betreft de ontwikkeling en evaluatie van verschillende methoden om taalkundige kennis zoals bijvoorbeeld vastgelegd in woordenboeken, morfologische regels en parallelle teksten, te combineren met statistische taalmodellen. Het idee is dat de zuiver statistische modellen hiermee zouden kunnen worden verbeterd of uitgebreid. In het bijzonder is gekeken naar de morfologische analyse van het Nederlands (het afbeelden van woordvormen op hun stam) en naar het zoeken in documenten die in een andere taal gesteld zijn als de zoekvraag (cross-linguaal zoeken). Voor het Nederlands zijn verschillende technieken voor morfologische normalisatie ("stemming") met elkaar vergeleken: een versie voor het Nederlands van het Porter-algoritme, een lexicale database met morfologische kennis (CELEX) en een meer patroongeörienteerde techniek gebaseerd op trigrammen en edit-afstand. Alledrie de technieken leveren een significante verbetering op van de effectiviteit van het zoeksysteem. Vervolgens is bekeken of een meer complex IR-model dat gebaseerd is op gewogen stemming en matching op woordvormniveau in plaats van stamniveau, leidt tot verbeterde resultaten. Dit bleek niet het geval te zijn. De voorlopige conclusie is dat het gunstiger is om te werken met een representatie op basis van woordstammen omdat de parameters van het taalmodel dan robuuster kunnen worden geschat. Voor het cross-linguaal zoeken is een vergelijking gemaakt van verschillende manieren waarop woord voor woord vertaling kan worden gekoppeld aan een IR-model gebaseerd op taalmodellen. De varianten waarin vertaling als een statistische component is geïntegreerd leverden de beste resultaten. Parallelle corpora die zijn opgebouwd uit webpagina's en hun vertalingen bleken uiterst geschikt voor het genereren van de benodigde statistische vertaalwoordenboeken. Deze methode van woordenboekgeneratie levert vaak ook een aantal geassocïeerde termen op als pseudo-vertaling. De geïntegreerde retrievalmodellen bleken hier op een effectievere

manier gebruik van te kunnen maken dan de vaak gebruikte techniek op basis van de synoniem-operator.

Het tweede onderwerp betreft de vraag of het mogelijk is om een IR-model gebaseerd op statistische taalmodellering te definïeren, dat zowel geschikt is voor 'topic tracking' als de 'ad hoc' zoektaak. Topic tracking verschilt van ad hoc zoeken doordat voor eerstgenoemde taak de statistische verdeling van retrievalscores voor verschillende zoekvragen vergelijkbaar moet zijn op een absolute schaal. Normaliter worden voor beide taken twee verschillende modellen gebruikt. In het proefschrift wordt een IR-model beschreven dat bruikbaar is voor beide taken. Het model is gebaseerd op de reductie in cross-entropie van een zoekvraagmodel gegeven een documentmodel in vergelijking met een achtergrondmodel. In meer informele termen: het IR-model bepaalt hoeveel beter het documentmodel de zoekvraag modelleert dan een neutraal model. De op cross-entropie gebaseerde aanpak is ook geschikt voor een transparante modellering van verschillende varianten voor cross-linguaal zoeken door de verschillende taalmodellen die in de formule een rol spelen te projecteren op één en dezelfde taal.

Het derde onderwerp dat behandeld wordt in het proefschrift is een meer algemeen probleem voor IR-onderzoekers, namelijk de statistische validatie van experimentele resultaten. Standaard tests voor statistische significantie worden besproken en de onderliggende aannames met betrekking tot de eigenschappen van de testdata worden gevalideerd op basis van gegevens van IR-experimenten. Dit resulteert in een aantal richtlijnen die bijdragen aan een verbeterd methodologisch kader. Het eerste deel van het proefschrift bevat bovendien een uitgebreide en up-to-date beschrijving van de 'state-of-the-art' op het gebied van retrievalmodellen en -technieken. Het kan gelezen kan worden als een inleiding op het vakgebied information retrieval.

Curriculum Vitae

Wessel Kraaij was born on may 14th, 1963 in Eindhoven. He finished secondary school (Gymnasium-B) in 1981 at the Eindhovens Protestants Lyceum and continued his studies at the Faculty of Electrical Engineering at the Eindhoven University of Technology. His master's thesis(1988) at the former Philips' Institute for Perception Research (IPO) concerned the design and experimental interactive evaluation of a system for making textual or voice annotations for a word-processor. The cognitive aspects of this research topic stimulated his interest in human-computer interaction in a broad sense. He started his professional career at the former institute for language technology and AI (ITK) of the University of Tilburg as a research-assistant, working on natural-language interfaces and dialogue systems. For a short period (1994-1995) he was affiliated as a research assistant at the Utrecht Institute of Linguistics (OTS) of Utrecht University. During this period he worked on a project geared towards improving text retrieval for Dutch using linguistic knowledge, which formed the starting point for this dissertation. In 1995, he accepted a position at TNO TPD in Delft (a research institute founded by the Dutch government), where he is currently working as a senior researcher and project manager in the department of Data-Interpretation. From November 1999 until July 2000, he was a visiting researcher at *Laboratoire Recherche Appliqueé en Linguistique Informatique (RALI)*, Université de Montréal.

His main research interests are language technology and multimedia retrieval. He (co-) authored about 35 papers for workshops, conferences and journals as well as several book chapters in the areas of cross language information retrieval, web retrieval, video retrieval, summarization, spoken document retrieval, topic detection and tracking, language modeling for IR and NLP enhanced IR. He is a regular reviewer for IR journals and program committee member of conferences in the area of information retrieval and natural language processing. Since 2003 he is co-coordinator of the NIST TRECVID benchmarking workshop on video retrieval.