# Improving Pseudo-Relevance Feedback: "Multiple Paths, Same Goal"

**B.Tech. Project Report**
**Stage 2**

Submitted in partial fulfillment of the requirements
for the degree of

**Bachelor of Technology**
**(Computer Science and Engineering)**

by

**Karthik Raman**
**Roll No: 06005003**

under the guidance of

**Dr. Pushpak Bhttacharya**
Indian Institute of Tecnology, Bombay
*and*
**Raghavendra Udupa**
Microsoft Research, India


Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

# Acknowledgements

I am extremely thankful to my guides Professor Pushpak Bhattacharya and Raghavendra Udupa(Microsoft Research, India) for their constant encouragement and motivation as well as their guidance and valuable inputs. I would also like to thank Dr. A. Kumaran(Microsoft Research, India)for his support and assistance. I also am grateful to my senior Abhjit Bhole whose Information Retrieval Engine and previous work was a platform for beginning much of my work. I would also like to thank Manoj Chinnakotla, with whom I have collabarated in research, during this period. I would also like to thank my friends Adith and Nikhil who have provided valuable inputs and encouraged me.

Finally, I would like to thank my parents for their support and encouragement throughout the duration of this project.

# Contents

## Abstract

A common problem in Information Retrieval(IR) is when queries are too short to completely convey the user's information need. A well-known technique to fix this is Query Expansion(QE), where relevant, informative terms are added to the query, to convey the missing intent. Pseudo-Relevance Feedback(PRF) is one of the popular methods to perform Query Expansion, and is widely accepted to improve retrieval performance. However it is also known to have flaws, which limit performance. In this thesis we propose novel techniques to address these. One of the problems faced by PRF is that of *Lexical and Semantic Non-Inclusion*: The expansion terms extracted by PRF are primarily based on co-occurrence relationships with the query terms, thus ignoring *synonyms* and *morphological variants*. Another problem is the sensitivity of PRF to the quality of the documents from which terms are extracted, which causes PRF to *lack robustness*.

In this thesis we propose a novel approach named **MultiPRF**(Multilingual PRF) , which tries to ameliorate both of the above problems, by harnessing *multilinguality*. Given a query in language $L_1$(called the *source language*) the key idea in MultiPRF is to take the *assistance* of another language $L_2$(called the *assisting language*). The query in $L_1$ is translated into $L_2$ and PRF performed on a corpus of language $L_2$ using the translated query, and the resultant feedback model translated back into $L_1$, and finally combined with the original feedback model of $L_1$ to perform the final ranking. We perform numerous experiments on the CLEF collections in widely-differing languages such as *English, French, Spanish, German, Dutch, Finnish and Hungarian*, and verified that MultiPRF not only consistently outperforms PRF, but is also significantly more robust. We also thoroughly analyze the results, to find that the gains in MultiPRF, are primarily due to discovery of both co-occurrence based conceptual terms, as well as lexically and semantically related terms. We also study the dependence of the system performance on the query translation quality and find the method to be robust to sub-optimal query translation quality. Lastly, we look into the performance of different source-assisting language pair performances, to discover that improvements are more pronounced in the cases where the two languages are closely related, as in the cases of the French-Spanish and German-Dutch pairs. We next look to extend the above method and propose another method **MultiAssistPRF** (Multiple Assisting Pseudo Relevance Feedback), where instead of using just a single assisting languages, two or more languages can be used to improve performance. Preliminary results indicate tangible benefits of the method, over using a single assisting language.

PRF is sensitive to the documents in the feedback set due to the presence of irrelevant documents. As part of this thesis we propose ways of identify irrelevant documents. We define the concept of $pseudo-irrelevant$ documents and propose a method to identify them. We then propose methods of using these documents to improve PRF, by finding *discriminative* terms. We also propose a method to identify the irrelevant documents in the feedback set by training a logistic classifier, which amongst its' features uses proximity to the pseudo-irrelevant distribution as a feature. We test the performance of this classifier on multiple collections, and find it to be able to satisfactorily distinguish relevant documents from irrelevant documents in the feedback set. We also propose a method to use the output of the classifier to improve PRF by eliminating documents which are thought to be irrelevant.

We also briefly discuss the discriminative approach to Information Retrieval and study extensions to current approaches, which incorporate Named Entities and Term Similarities. We also propose a principled manner of incorporating PRF into this framework using local re-ranking. We also explore another approach to query expansion, which relies on a probabilistic thesaurus. Since probabilistic thesaurus are very rare to find and difficult to create, we discuss an approach to learn a thesaurus by performing a random walk across terms in different languages. Results show the learned thesaurus to be sensible, and good performance when used for query expansion. We conclude by relating all the different methods proposed in this thesis, and explore the possibility of combining some of these methods.

# Chapter 1

# Introduction

The central problem in **Information Retrieval(IR)** is that of satisfying the information need of a user, who typically uses a short (2-3 words typically) and often ambiguous query to convey his intent. The task of any IR system is to rank documents, from a large collection, as a list, with the objective of getting the relevant documents to the top of the list. Thus an ideal system would rank documents based on their *true relevance* to the query. Since the true relevance of a document is unknown, the challenge is to use the only piece of available information: *the given query*, to *estimate* the relevance of a document. This problem is further complicated due to natural language phenomena like *morphological variations*, *polysemy* and *synonymy*.

Over the years different systems have modeled the query differently, with varying degrees of success. The vector space model [Sal71] was one of the first successful frameworks, in which queries and documents were treated as vectors in a term-space (with each dimension corresponding to a different term) and documents were ranked based on their cosine-similarity to the query. However due to the lack of a strong theoretical foundations, this was soon replaced by the theoretically-sound Binary Independence Retrieval (BIR) Model as proposed by Stephen Robertson et. al [RSJ88]. In this framework, given a query, the task of IR was likened to a binary classification problem, i.e. classifying documents as either relevant or irrelevant. A document $D$ is ranked based upon its' odds of being observed in the relevant class i.e. $P(D|R)/P(D/R')$, where $R$ is the relevance class and $R'$ is the irrelevant class. However since we are not explicitly given the Relevant or Irrelevant classes, the probabilities $P(D|R)$ and $P(D/R')$ become difficult to estimate. Lavrenko and Croft [LC01a] went some way towards fixing this by their relevance-based language models, which gave a non-heuristic approximation to the relevance model.

Another significant area of research has been in the field of language models [PC98, SC99, Zha] which was initially proposed by Ponte and Croft. Here both documents and queries are treated as Language Models, which are simply probability distributions over the set of terms in the vocabulary. Here instead of trying to explicitly derive the relevance models, the probability of the query being generated from a document's language model i.e. $P(Q|D)$ is found, and is further used as an indicator of the relevance of the document to rank the documents. To compute this easily, query-term independence is assumed, thus leading to the simplification of the expression:

$$P(Q|D) = P(q_1, q_2, ...q_k|D) = \prod_{i=1}^{i=k} P(q_i|D) \tag{1.1}$$

Due to their strong theoretical grounding, their modeling capabilities and simplicity *language models* have proven to be very powerful. Research has also been done on methods to improve these models using techniques such as positional language models [LZ09], proximity-based language models [ZY09] and smoothing [ZL04].

However none of these models directly solve the problem of queries being short, with incomplete user intent. Thus using the given query terms alone, is not good enough to get a good idea of the user's intent (and hence the relevant documents). Thus to convey the complete intent and information need of the user, we append certain relevant terms to the query. This is called **Query Expansion** [Eft96]. Since shorter queries tend be more ambiguous, appending keywords to the query, clarifies the intent and helps bring the query closer to the documents present in the corpus. Query Expansion can be done either automatically or by involving the user in the loop. Furthermore query expansion techniques can be classified as either "global" methods or "local" methods [MRS08].

*Global Methods* refer to methods which utilize a global resource such as a thesaurus, or corpus-wide statistics. Since these are not specific to the given query, they are called global methods. Global methods also include spelling corrections, refining the query using synonyms from a thesaurus and other similar methods. Synonym-based Query Expansion, using a thesaurus, is one of the well-studied methods, with theoretical justifications. The underlying concept in synonym-based expansion is: While a document may not include the given query terms, it may instead contain synonyms of the query terms. For example consider the query on "Mountain Lions". For this query a document talking about "cougars" is also relevant, as the two terms are synonyms of each other (an example from Hindi is where the query could be on *pani*, but the document instead contains the terms *jal* or *neer*). Another global method, involves suggesting possible additional query terms to the user. For example the query string "apple" is ambiguous as it could refer to one of multiple things: Apple as in the company or the Apple i-pods, the Apple computers or simply the fruit Apple. Thus suggesting possible expansions and asking the user to clarify their intent is an example of global query expansion. Similarly modern day search engines suggest possible spelling corrections when there is a perceived error in the spelling of the query terms, which is another example of query reformulation.

*Local Query-Expansion Methods* refer to methods where the query is modified based on some query-specific statistics. They generally use knowledge from the top documents, from an initial retrieval, that appear to match the query. This could be either automatic (i.e. **Pseudo-Relevance Feedback** or user-driven (i.e. **Relevance Feedback**).

*Relevance Feedback* is a user-driven process, wherein the user clarifies his intent using the top few retrieved documents. The user can indicate if he/she think a particular retrieved document, is either relevant or irrelevant. The retrieval engine then uses this "feedback" and suitably modifies the query, based on what it believes to be the user's true intent given the feedback it received. Modern-day search engines provide for such feedback in the form of the *More Like This* and *Hide* options. While these are examples of explicit feedback, where the user explicitly signals their

intent, feedback can also be implicit. If a user skips past the first few results displayed, and instead chooses to view a document retrieved at a later position, then there is a good chance that the top few documents, which were skipped, did not match the user's intent. There has been work on this done by Joachims et.al. using clickthrough data, to get such feedback [JGP+05, JR07]. This kind of feedback is much easier to obtain (using query-logs) and is generally reliable. There has also bee recent work on using this implicit feedback, to compare different ranking functions as well [YGJ+10].

In some scenarios, user feedback is hard to obtain or unreliable. In such cases a third kind of feedback is used called *Pseudo-Relevance Feedback*(PRF) [BSAS94, XC00, MSB98]. This is also known as *Blind Relevance Feedback*, as the user is completely removed from the process. In this method, since no knowledge of relevance/irrelevance is provided, the top $k$ documents retrieved by a first-pass baseline retrieval are assumed to be relevant. Based on this assumption, the query is expanded using the selected set of top $k$ documents, which is also referred to as the *feedback set*. The terms in the feedback document set are analyzed to choose the most distinguishing set of terms that characterize the feedback documents and as a result the "assumed" relevance of a document to the given query. Query refinement is done by adding the terms obtained through PRF, along with their weights, to the actual query. This simple framework, is well-accepted to improve retrieval performance in most IR systems.

Despite this, PRF is known to have flaws in its' framework, namely:

1. *Lexical and Semantic Non-Inclusion*: The type of term associations obtained for query expansion is restricted to co-occurrence based relationships in the feedback documents, and thus other types of term associations such as lexical and semantic relations (morphological variants, synonyms) are not explicitly captured. As noted in the example given for synonym-based expansion, lexically and semantically related terms can be critical to improving performance as many of these terms cannot be obtained via co-occurrence based relationships. For the "mountain lion" example, it is unlikely that we will find "cougar" co-occurring with "mountain lion" and may thus be forfeiting significant performance benefits.

2. *Instability and Lack of Robustness*: The inherent assumption in PRF (*i.e.*, relevance of top $k$ documents) is too strong and almost never holds in practice. This causes **Topic Drift**, *i.e.* the concept behind the expanded query tends to drift away from the intended meaning of the original query. We highlight how this leads to severe performance decrease in PRF techniques, causes loss of stability (*i.e.* performance drops for some queries after using query expansion on them), a lack of robustness and heightened sensitivity to performance of initial retrieval algorithm.

Both of the above problem have been tackled, to an extent, but separately. While approaches similar to synonym-based expansion solve the first problem, using another (preferably larger) collection, such as Wikipedia helps ameliorate the second problem. However we try to come up with a framework that solves both of the problems simultaneously. In this thesis, we propose a novel approach called **Multilingual Pseudo-Relevance Feedback (MultiPRF)** [CRB10a, CRB10b] which overcomes both the above limitations of PRF. The key idea here is the use of resources in another language to "assist" in retrieval, *i.e.* we do so by taking the help of a different language called the *assisting language*. In MultiPRF, given a query in *source language* $L_1$, the query is

automatically translated into a query in the assisting language $L_2$. In the next step, PRF is performed on a collection in the assisting language using the translated query, and the resultant terms obtained are translated back into $L_1$ using a probabilistic bi-lingual dictionary. The translated feedback model, is then combined with the original feedback model of $L_1$ to obtain the final model, which is then used to re-rank the corpus. Experiments on standard CLEF [BP04] collections in languages from different linguistic families and widely different characteristics, namely *Dutch, German, English, French, Spanish, Finnish* and *Hungarian* show that MultiPRF not only achieves significant performance improvement over monolingual PRF, but is also more robust. The effect is all the more pronounced when the source language is assisted by a closely related language- for example, the French-Spanish and Dutch-German pairs. We study this effect more closely and try to postulate the possible reasons for this phenomenon.

A thorough qualitative analysis of the results is performed, and query-wise analysis is also done. To provide greater insight, queries with large improvements as well as queries whose performance decreases on using MultiPRF, are analyzed in depth, and the expansion terms obtained by MultiPRF for these queries also studied. These studies reveal that the second language helps in obtaining both co-occurrence based conceptual terms as well as lexically and semantically related terms, unlike standard PRF and thus rectifying the first flaw of PRF. Additionally, since MultiPRF uses PRF in two collections of different languages, it is expected to be more robust as well, since this is similar to a risk-minimization step. This can be understood by realizing that the probability of a query failing in two languages is always less than it failing in one, thus leading to greater robustness. Experiments show that it is a combination of both of these factors which leads to the large gains seen, and not just one of them.

The dependence of the system performance on the query translation quality is also studied, by using different query translation systems with varying degrees of translation accuracy. This leads us to the observation that the method is robust to sub-optimal query translation quality and outperforms standard PRF techniques even with poor query translation accuracy. The MultiPRF method uses a parameter to control the relative weightage of the translation feedback mode, the original feedback model and the query. We study the effect of varying the value of this parameter and the corresponding change in the performance, and for a small range of parameter values, find the method to be relatively insensitive to the parameter value.

We next study a natural extension of the above method, where instead of using just a single assisting language, we combine the evidence from multiple languages. The method, called **Multi-AssistPRF** (Multiple Assisting Pseudo Relevance Feedback) uses the translated feedback models from two or more assisting languages. Two orthogonally different ways of combining these models are studied as well. While in one method, all the different models are simply interpolated with equal weightage, in the other method only a few of the models are chosen based on a certain criterion. Both of these methods can be thought of as special cases of a general method where a weighted interpolation of the different models is performed. Finally, we see preliminary experiments showing tangible benefits of this method over using a single assisting language.

Inspired by the above method, and the accumulation of synonyms and morphological variants by going to a different language we propose another novel approach to perform query expansion. The method, which is in the language modeling framework like the previous one, is not a PRF technique, and depends only on a probabilistic thesaurus. Probabilistic thesauri are a very potent resource, and find applications in many fields of NLP. However in practice, these are very rare as

they are extremely difficult to create. In this thesis we propose a method to create such a thesaurus given probabilistic bilingual dictionaries (which are easily available and simple to create) between two the source language $L_1$ and a *friend language* $L_2$. In this method, we perform a random walk across terms from one language to terms in another language using the edge weights as given in the probabilistic dictionaries. Preliminary results show the thesauri learned by this method to be sensible. When this thesaurus is used for query expansion, the performance is found to be comparable to synonym-based expansion systems.

As mentioned earlier, the PRF assumption is too strong and holds only in rare cases. In practice the presence of irrelevant documents leads to performance drops. We study how the presence of irrelevant documents in the top $k$, adversely affects popular PRF methods, and limits the performance of these methods. We also see that performance can decrease due to the presence of irrelevant documents outside the top $k$ as well. This happens due to these irrelevant documents being ranked above some other relevant documents. We observe the benefits possible in retrieval performance if these irrelevant documents are identified, and eliminated from the list.

Motivated by the potential benefits in performance, we propose a method to identify high-scoring irrelevant documents outside the top $k$. We show how our method can accurately identify these irrelevant documents across different collections. We introduce the concept of *pseudo-irrelevance* [RUBB10] analogous to the concept of pseudo-relevant documents. We show how the set of pseudo-irrelevant documents provide a good, accurate approximation to the set of high-scoring irrelevant documents.

We next look at how these pseudo-irrelevant documents, can be utilized to improve the performance of PRF. We show how the set of pseudo-irrelevant documents can be used to identify *good expansion terms*. It finds terms which help discriminate between the "assumed relevant" documents, *i.e.* pseudo-relevant documents and the "largely irrelevant" set of pseudo-irrelevant documents. We learn a logistic classifier to discriminate between these two sets, using terms as features, and based on the feature weights which are learnt, we extract good expansion terms. We display the improvements obtained by our methods across different collections, as well as propose some ideas which could lead to further improvements.

Next we tackle the biggest challenge: Identifying the irrelevant documents inside the top $k$. We propose a few discriminative features which can help us distinguish between the relevant documents and the irrelevant documents in the feedback set. We also observe how the pseudo-irrelevant documents are more similar to the irrelevant documents in the feedback set as compared to the relevant documents in the feedback set, and thus can be used as one of the discriminative features, to identify some of the irrelevant documents from the feedback set. We create a linear classifier, using the proposed features, and try to classify documents in the feedback set as either relevant or irrelevant. We observe how our approach is able to identify many of the irrelevant documents from the relevant documents. We also discuss why this task is especially challenging, and why there is very less hope to do better. We then discuss two orthogonal methods of using these predictions to alter PRF. The first approach defines a fuzzy measure of relevance associated with each document in the feedback set. We next incorporate this measure into a simple document-weighting scheme, analogous to the popular term-weighting schemes used in IR, and describe how these document-weighting schemes can be combined with some of the common PRF techniques. In the second approach we take more drastic action, by considering all documents below a certain threshold to be irrelevant and eliminate them from the feedback set. We then discuss the pros and cons of each

of these approaches.

We also motivate this line of work by observing how standard PRF techniques benefit from having more relevant documents within the PRF set, and thus propose an extension to the above ideas based on replacing identified irrelevant documents (from the top $k$) with *high-confidence relevant* documents, i.e. documents just outside the top $k$ which appear to be relevant with a high degree of confidence. These high-confidence documents are again obtained by using a linear classifier with certain new features as compared to before. Finally we provide a framework to combine the above ideas, as a single system, that could potentially lead to large performance improvements and robustness.

While language models fall within the framework of Generative Models, in recent times, a different class of models has received significant interest in Web Search. These are the **Discriminative Models**. Recent work by Nallapati [Nal04] extended the domain of discriminative models to information retrieval. Joachims et. al. proposed RankSVM [Joa02], which put the problem of ranking within the discriminative framework, and provided a theoretically strong foundation to build on. There has been lot of work done since in this area, but they have tended to focus on optimizing the Rankings for different measures of ranking such as *AUC*, *MAP* and *NDCG* [Joa05, YFRJ07, CKSB08]. Unlike Language Models, discriminative models do not require finding a probability distribution over terms. We try to study different extensions of information retrieval techniques to discriminative models. We firstly try to extend the concept of Query Expansion and PRF to the discriminative framework. Initial experiments suggest that although the standard formulations of PRF (in terms of a language model) lead to improvements over baseline methods, they still do not perform as well as their language model counterparts. Hence we investigate a more natural formulation of PRF in the discriminative context, wherein we use the concept of local re-ranking, *i.e.* re-ranking only the top 1000 documents.

We conclude by relating all the different methods proposed in this thesis, and drawing comparisons amongst them. Since some of these models are strongly related while others tackle a different problem, we also explore the possibility of combining some of these methods to do better.

# Chapter 2

# Query Expansion and Pseudo-Relevance Feedback (PRF)

In this section we review Query Expansion and Pseudo Relevance Feedback and other related topics. Query Expansion (QE) involves appending appropriate terms to the original query to help disambiguate the user intent. Pseudo-Relevance Feedback(PRF) is a popular method used for automatic Query Expansion, wherein the top-ranked documents are used to extract expansion terms, by making the assumption that the top $k$ documents are relevant. However this assumption rarely holds in practice, thus affecting the performance of PRF.

## 2.1 Query Expansion

In Information Retrieval and Web Search, queries from users tend to be short and ambiguous. Although they have some specific information need in mind they tend to express this in very few words. It is well known that the longer this expression of intent *i.e* the query, the less ambiguity involved. Hence to rectify this problem, *Query Expansion* [Eft96] was proposed. They key idea here is that,since shorter queries tend be more ambiguous, appending keywords to the query, would help in clarifying the intent of the user and thus satisfy the information need of the user.

Query Expansion methods can be either classified as being automatic or manual *i.e.* involving the user in the loop. Furthermore query expansion techniques can also be classified as either being "global" or "local".

1. *Global Methods*: Those Query-Expansion methods which make use of some global resource such as a thesaurus, or corpus-wide statistics. Unlike local methods, they do not any specific processing for a particular query. Examples of Global methods include spelling corrections and refining the query using synonyms from a thesaurus. The latter *i.e.* Synonym-based Query Expansion is one of the well-studied "global" methods of Query Expansion. The intuition behind this is that the query may be expressed using terms, which are not frequent

in the corpus. Rather, synonyms of these terms are what occur frequently in the corpus. Thus unless we add these terms to the query, we cannot achieve high recall for such queries. This is referred to as *synonymy, i.e.* the phenomena where the same concept can be described using different words. Another global method, involves suggesting possible additional query terms to the user. Modern day search engines such as Google offer a "Do You Mean" option as well, which amongst other things, suggests possible spelling corrections when there is a perceived error in the spelling of the query terms. This is an example of global query reformulation.

2. *Local Methods*: Query Expansion Methods which depend on the given query and/or some statistics computed for this query, are clubbed under Local Methods. A popular class of Local Expansion Methods use knowledge from the top documents of an initial retrieval. Relevance Feedback and its' variations are an example of such methods.

## 2.2 What is PRF?

Pseudo-Relevance Feedback (or Blind Relevance Feedback) is an automatic method of performing query expansion. Unlike Relevance Feedback, here there are no relevant documents provided. Since the method is meant to be automatic and without user interaction, in PRF the top $k$ documents (retrieved by a first-pass retrieval) are assumed to be relevant. Based on this assumption the objective is to find suitable terms from these documents, which are referred to as the *feedback set*, for expanding the query. The concept behind the method is that if the assumption holds and the top $k$ documents are relevant, then the expansion terms obtained from these documents can help retrieve other relevant documents as well as go some way towards clarifying the user intent, by disambiguating some of the query terms.

## 2.3 PRF and Language Models

PRF techniques have generally been studied in the Language Model setting. PRF techniques are generally of two kinds : either assign weights directly to a set of terms, or find the top few terms and add them to the query. These terms are obtained from the top $k$ documents which are called the *Feedback Documents*. In both techniques eventually a Language Model is created which is called the *Feedback Model*. This feedback model is then interpolated with the original query Language Model to obtain a new interpolated query model, which is used for the second-pass retrieval.

Hence in this framework of PRF using Language Models, the problem of interest is choosing the appropriate terms to expand the query, and what weight should each of them be assigned. Early PRF techniques focused on getting high IDF terms. There have been many other heuristics proposed for sorting and choosing top terms. There has been some recent work focusing on estimating the feedback model directly from the feedback documents itself [ZL01]. Though these methods may use different criteria for selecting the top terms, the underlying principle common to all of them, is that they look for terms which separate the *pseudo-relevant* feedback documents from the other documents in the collection. In other words they look for terms which help in discriminating the feedback documents from the other documents in the collection. This is a very powerful concept,

and something which we explicitly use in our methods as proposed in Chapter 9. Another common aspect in all these models is that they interpolate the final feedback model with the initial query model. This is because using the feedback model alone is not sufficient, as then the context/intent of the query may be lost.

## 2.4 Concept of Risk-Minimization

Risk is a well-known concept in the fields of economics and finance. Zhai and Lafferty introduced the concept of risk in the field of Information Retrieval. They proposed a method of PRF [ZL01] which naturally incorporated the concept of risk-minimization. They considered the *risk* involved if they only chose a few set of terms for query expansion. Consider a PRF technique which sorted the terms in the feedback documents (as per some measure e.g. IDF) and only chose the top few terms. If there was a particularly important term in the relevant documents, which was not chosen for expansion, then retrieval performance could be affected. Hence ignoring a term has a "risk" associated with it.

Based on this principle of risk they proposed a model, in which rather than selecting only a few set of terms, they use a large number of terms for expansion. This way they try to minimize the risk involved in query expansion. Choosing a large set of terms too could lead to very bad performance as a lot of noisy non-topical terms may be chosen. Hence it is important to distinguish between the seemingly important terms, and the terms which seem to be noisy. Thus to get the right balance between performance and risk-minimization they proposed a method to assign weights to each of the expansion terms, which were combined into the feedback language model. Thus they managed to extract the feedback model directly from the feedback documents , while maintaining a good balance between risk-minimization and performance. Thus this model is one of the most principled and well-accepted models of PRF, due to the it's robustness and performance. The model is further discussed in Section 3.2.

## 2.5 Pseudo-Relevance Feedback in Discriminative Models

As opposed to the Generative Models such as Language Modelling, there has been a lot of work recently on Discriminative Models in Information Retrieval. However there has been no documented work of Pseudo-Relevance Feedback in the Discriminative Model setting in Information Retrieval. This is rather surprising as discriminative models offer more freedom for modelling, due to the varied forms the features of the model can take. While Nallapati [Nal04] proposed a set of features for Discriminative Models, there has been no work of extending these concepts to PRF; nor have there been any PRF features proposed for usage in the discriminative context. We explore this area further in Chapter 13.

## 2.6 Topic Drift

Using PRF allows for the possibility of **Topic Drift**. This is the biggest problem associated with it. Topic Drift indicates when the topic/underlying intent of the expanded query has drifted/moved away from the underlying intent of the original query. Thus if there was a query *Drugs in Soccer*, and the feedback documents talked about "Maradona and his use of drugs" then there would be a drift of the query from "Drugs in Soccer" to "Maradona". Hence the kind of documents in the feedback set, and their content can influence heavily the intent of the feedback model, and how closely it matches the intent of the original query.

This becomes a much more serious issue when the feedback set has a lot of irrelevant documents within it. Since the irrelevant documents, have an underlying topic which is different from the intended meaning of the query, picking terms from such documents may cause the topic of the expanded query to be very different from that intended. This is one of the key reasons which limits the performance of PRF. We shall see how PRF is affected by presence of irrelevant documents in the feedback set in Chapter 10.

# Chapter 3

# Related Work

In this chapter we look more closely into existing research work, which are related to our work, or have inspired our work. We first look at the popular technique of Language Modeling, and the smoothing methods employed to improve performance of Language Models. Next we look at Model-Based Feedback, which is one of the most principled and popular PRF techniques as proposed by Zhai and Lafferty.

We then look into existing work, which is similar to MultiPRF. We look at work which tries to use multiple collections to help improve robustness in PRF. We also look into an approaches which have tried to use resources from other languages as well. Finally, we look at some work, which is similar in essence to MultiPRF but has some differences. We compare our method with theirs and try to analyze the pros and cons of each.

Since one of the focus points of this thesis is the topic of irrelevant documents, we delve into existing work which also focuses on irrelevant data. The Rocchio model is one of the oldest models which is used for Relevance Feedback. We study it to see how it handles negative feedback. Next we look at some of the recent work of Zhang et.al. [ZHS09] which also involved improve Pseudo-Relevance Feedback using Irrelevance Data. Though they do not discuss the identification of irrelevance data, they propose a method to improve the feedback model, using irrelevance data. Finally, we look into some work by Zhai and Lafferty [LZ01], which inspired our work pertaining to the random walk model for creating a probabilistic thesaurus.

## 3.1 Study of Language Models and Smoothing Methods

Language Models were proposed by Ponte and Croft [PC98]. Language Models are essentially probability distributions over a set of terms. Given a set of terms $t_1, t_2, ...., t_k$, a language model $L$ over these set of terms is basically a Probability Distribution $P(t_i|L)$ for $i = 1, 2, ..., n$, where:

$P(t_1|L) + P(t_2|L) + ....P(t_k|L) = \sum_{i=1}^{i=k} P(t_i|L) = 1.$

The probability of a term $t_i$ i.e. $P(t_i|L)$ corresponds to the generation probability of the term $t_i$ as per the Language Model $L$. Ponte and Croft introduced Language Modeling in IR. In IR the objective is to find documents which match the user's query. In a probabilistic framework, this can be given as $P(D|Q)$, i.e. Probability of document $D$ satisfying the user's requirement given the query $Q$. Since we want the documents which match the query the closest, this amounts to ranking as per decreasing $P(D|Q)$. However since the set of documents in a corpus tends to be large, with each document being much longer than the query itself, directly computing $P(D|Q)$ is not feasible. Hence we use Bayes' Rule here:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}$$

$P(D)$ is a prior over the documents which is generally assumed to be a uniform prior(For web search, this is obtained from the PageRank algorithm). Similarly $P(Q)$ is constant for a given query. Hence the problem of ranking amounts to ranking documents as per decreasing value of $P(Q|D)$. To model $P(Q|D)$ we use the framework of language models.

First we model the query as a language model over the set of query terms : $q_1, q_2, ....q_k$. Similarly a document can be modeled as a language model, known as the *document model*. The unsmoothed document model is simply a language model over the set of terms appearing in the document. This can simply be obtained as an MLE estimate, i.e.

$$P(t_i|D) = \frac{count(t_i,D)}{|D|},$$

where $count(t_i, D)$ is the number of occurrences of term $t_i$ in the document $D$, while $|D|$ is the length of the document, i.e. the total number of lexicons appearing in it. The query language model can also be computed in a similar manner.

However since a document may not contain all the query terms, using the unsmoothed model was shown to have poor performance and hence instead we use a smoothed document model, which is a linear interpolation of the unsmoothed document model $P(w|D)$, and the collection model $P(w|C)$:

$$P'(w|D) = (1 - \alpha_D)P(w|D) + \alpha_D P(w|C) \tag{3.1}$$

Now to find $P(Q|D)$ we use a term-independence assumption and write it as:

$$P(Q|D) = \prod_{q_i \in Q} P(q_i|D)^{count(q_i,Q)}$$

Ranking over decreasing $P(Q|D)$ amounts to ranking over decreasing $\log P(Q|D)$.

$$\log P(Q|D) = \sum_{q_i \in Q} count(q_i, Q) \log P(q_i|D) = n_Q(\sum_{q_i \in Q} P(q_i|Q) \log P(q_i|D))$$

where $n_Q$ is number of query terms. Since we are interested in ranking the documents we ignore the $n_Q$ term. To justify why we have used this form of $P(Q|D)$ we show that ranking as per decreasing $(\sum_{q_i \in Q} P(q_i|Q) \log P(q_i|D))$, is an accurate measure of similarity between the two distribution $P(w|Q)$ and $P(w|Q)$.

To do so we first subtract $\sum_{q_i \in Q} P(q_i|Q) \log P(q_i|Q)$. Hence we are ranking based on the value of decreasing :

$$\sum_{q_i \in Q} P(q_i|Q) \log \frac{P(q_i|D)}{P(q_i|Q)}$$

or on the increasing value of :

$$\sum_{q_i \in Q} P(q_i|Q) \log \frac{P(q_i|Q)}{P(q_i|D)}$$

This can be seen to be similar to the Kullback-Leibler(KL) Divergence between the two probability distributions $P(w|Q)$ and $P(w|D)$. The KL-Divergence [KL] is a very popular formulation of capturing the distance between two probability distributions. It is always $\geq 0$ and is $= 0$ only when the two probability distributions are the same. Hence ranking as per decreasing $(\sum_{q_i \in Q} P(q_i|Q) \log P(q_i|D))$ is equivalent to ranking as per increased divergence between the query language model and document model.

However instead of using the document model directly, it is smoothed first as given in Equation 3.1. There have been many studies over which smoothing techniques to use. Zhai [Zha] studied different smoothing techniques and established that performance can vary a lot depending on smoothing. He proposed a two-stage smoothing which is given as:

$$\alpha_D = \frac{(1-\lambda)|D|+\mu}{|D|+\mu}$$

This is inspired by the Jelinek-Mercer and Dirichlet smoothing techniques, and performs consistently over queries of different lengths. Hence using this value of $\alpha_D$ we get that the documents in a collection are ranked as per the decreasing value of :

$$\sum_{q_i \in Q} P(q_i|Q) \log \left(\alpha_D P(q_i|C) + (1 - \alpha_D) P(q_i|D)\right) \tag{3.2}$$

In our work, this model of ranking is used as the first-stage retrieval (unless explicitly mentioned otherwise). We refer to this model as the *LM-Retrieval*. We rank documents as per decreasing value of the quantity in equation 3.2, with two-stage smoothing. Results of the performance of the method will be provided in later chapters.

## 3.2 Model - Based Feedback

Zhai and Lafferty proposed a new approach to pseudo-relevance feedback wherein they "treat feedback as updating the query language model based on the extra evidence carried by the feedback documents". They justified their approach, which they coined *Model-Based Feedback*, by claiming that it is a natural extension to the language-modeling framework. They try to find a feedback model $\theta_F$, which they then interpolate with the initial query model $\theta_Q$ i.e.

$$\theta_{Q'} = \alpha\theta_F + (1-\alpha)\theta_Q$$

where $\alpha$ is the interpolation ratio. They provided a novel method of obtaining the feedback model $\theta_F$ using a Generative Model. In their formulation they try to remove the collection noise(background noise), using the collection language model, and thus obtain a cleaned topical model from the feedback documents. They proposed to find a generative model of the feedback documents, and assumed that the set of feedback documents $F = d_1, d_2, ...d_k$ are generated from the probabilistic model $P(F|\theta)$. They made a term-independence assumption and using this the log likelihood expression becomes:

$$\log P(F|\theta) = \sum_i \sum_w c(w, d_i) \log\left((1-\lambda)P(w|\theta) + \lambda P(w|C)\right)$$

where $P(w|C)$ is the collection model. Here we want to find that feedback model $\theta_F$ for which the log-likelihood is maximized. Here $\lambda$ is set to an empirical value and $\theta$ is estimated using the well-known EM algorithm. The EM update equations are:-

$$t_w^{(n)} = \frac{(1-\lambda)P^{(n)}(w|\theta)}{(1-\lambda)P^{(n)}(w|\theta)+\lambda P(w|C)}$$

$$p^{(n+1)}(w|\theta) = \frac{\sum_{i=1}^{i=k} c(w,d_i)t_w^{(n)}}{\sum_j \sum_{i=1}^{i=k} c(w_j,d_i)t_{w_j}^{(n)}}$$

After obtaining this model they prune some of the terms, which have very low probability values, to obtain the final feedback model which is then interpolated with the initial query model, and used to rank documents based on the score as per the KL-Divergence between the document model and the new expanded query model.

Due to its' principles framework, couples with strong performance, Model-Based Feedback is used as the benchmark relevance feedback technique for comparison with our methods, for the rest of this thesis. We refer to this model as the *MBF-Retrieval*. Results of the performance of the method on different collections and parameters used for the experiments will be provided in later chapters.

## 3.3 MultiPRF: Related Work

In this section we discuss some of the work which are related to our MultiPRF approach. Though this is a novel idea, with no direct previous work on the subject, there has been related work, in some other domains such as *Word-Sense Disambiguation* (WSD) [DIS91, KSKB09], which inspired some of the ideas in this work.

PRF has been successfully applied in various IR frameworks like vector space models, probabilistic IR and language modeling [BSAS94, JWR00, LC01b, ZL01]. Several approaches have been proposed to improve the performance and robustness of PRF. Some of the popular representative techniques which attempt to address these shortcomings of PRF are:

1. Refining the feedback document set [MSB98, SMK05]
2. Refining the terms obtained through PRF by selecting good expansion terms [CNGR08, UBB09]
3. Using selective query expansion [ACR04, CTZC04]
4. Varying the importance of documents in the feedback set [TZ06]

Although all these approaches, were shown to improve PRF, they do not satisfactorily fix the problem of lack of robustness. Another direction of work, often reported in the TREC Robust Track, explicitly tries to solve this problem, by using a large external collection like Wikipedia or the Web as a source of expansion terms [XJW09, Voo06]. The intuition behind the above approach is that if the query does not have many relevant documents in the collection then any improvements in the modeling of PRF is bound to perform poorly due to query drift. Hence by using a large collection such as Wikipedia, the number of relevant documents increases, thus making it more likely for PRF to find good expansion terms, in these larger corpora.

As already highlighted, PRF suffers from the problem of Lexical and Semantic Non-Inclusion. Several approaches have been proposed for including different types of lexically and semantically related terms during query expansion. Voorhees *et. al.* [Voo94] use Wordnet for query expansion and report negative results. Recently, random walk models [LZ01, CTC05] have been used to learn a rich set of term level associations by combining evidence from various kinds of information sources like WordNet, Web, co-occurrence relationships data, morphological variants data *etc.* Metzler *et. al.* [MC07] propose a feature based approach called *latent concept expansion* to model term dependencies. However although these approaches, solve the problem to an extent, they do not improve, and in some cases actually worsen, the robustness of PRF.

All the above mentioned approaches use the resources available *within* the language to improve the performance of PRF. However, we make use of a *second language* to improve the performance of PRF. Our proposed approach is especially attractive in the case of resource-constrained languages where the original retrieval is bad due to poor coverage of the collection and/or inherent complexity of query processing (for example *term conflation*) in those languages.

As mentioned before, the idea of using one language to improve the accuracy in another language has been successfully tried for the problem of Word Sense Disambiguation (WSD) [DIS91, KSKB09], where approaches like parameter projection allow for the parameters learned in one language to be "transferred" (*transfer learning*) to another language.

There are some other work, which share common ideas with the MultiPRF approach. In [JJJW99], Jourlin et. al. use *parallel blind relevance feedback*, *i.e.* they use blind relevance feedback on a larger, more reliable parallel corpus, to improve retrieval performance on imperfect transcriptions of speech. Another related idea is that by Xu et. al. [XFW02], where they learn a statistical thesaurus by using the probabilistic bilingual dictionaries of Arabic to English and English to Arabic. We use this idea as a baseline, for synonym based expansion, when we have to compare with our method. This idea can also be thought of a 1-step random walk from the English to Arabic and back to the English terms, and hence is strongly related as well to our idea for creating a statistical thesaurus. There is also work by Meij et. al [MTdK09], which tries to expand a query in a different language using language models for domain-specific retrieval, but in a very different setting. Since our method uses a comparable corpus [1] in the assisting language, it can be likened to work by Talvensaari et. al [TLJ+07] who used comparable corpora for Cross-Lingual Information Retrieval. Other work pertaining to document alignment in comparable corpora, such as [BS98, LCC02], also share certain common themes with our approach.

### 3.3.1 Work By Gao et. al.

The work that comes closest to our approach is some recent work by Gao *et. al.* [GBZ08], which uses English to improve the performance over a subset of Chinese queries whose translations in English are unambiguous. They too are motivated by the strong developments in retrieval for English, such as Page Rank, Click-Through Data and Taxonomies. Since this data is unavailable in other languages they propose to use features from English, to improve performance on a small subset of Chinese Queries.

They first translate the Chinese query into English (easy, since the query is unambiguous).They next find inter-document similarities across languages, i.e. English documents similar to Chinese documents and vice-versa. They then combine all this information to improve the ranking performance, by using an SVM formulation for the problem.

In their approach, the computation of cross language document level similarities between English and Chinese documents is done using a bi-lingual dictionary. However, cross language document similarity measurement is in itself known to be an equally hard problem especially without using parallel or comparable corpora [DLLL97]. Moreover, the scale of their experimentation is quite small and they demonstrate their approach only on a small class of queries in a single language. Also their settings for the experiments are restrictive since they impose conditions on the kinds of Chinese Queries.

[1] By a comparable corpus, we mean the same concepts should be covered in both corpora. They do need to have similar documents, and hence such corpora are relatively easier to obtain. For example, newswire corpora from the same time frame.

## 3.4 Irrelevance Based Experiments: Related Work

In this section, we discuss some of the popular methods, which inspired us, on our work pertaining to the experiments in Chapters 8, 9, 10, 11 and 12. In particular we discuss the Rocchio Algorithm, which was one of the first algorithms proposed, to perform feedback. We also discuss other methods used to perform negative feedback, including the Distribution Separation Method proposed by Zhang et. al.

### 3.4.1 Rocchio Algorithm for Relevance Feedback

The Rocchio Algorithm [Roc71, MRS08] as proposed by Rocchio is one of the classic algorithms for implementing relevance feedback. It uses the vector-space model to incorporate relevance feedback. Given a set of relevant documents and a set of irrelevant documents, it tries to find the optimal query vector $\vec{q}$. to maximize similarity with the relevant documents, while minimizing the similarity with the irrelevant documents. We represent the set of given relevant documents as $D_R$ and the given irrelevant documents as $D_{NR}$; the set of all relevant documents as $C_R$ and all irrelevant documents as $C_{NR}$. Using this notation we get:

$$\vec{q}_{opt} = max_{\vec{q}} \left(sim(\vec{q}, C_R) - sim(\vec{q}, C_{NR})\right)$$

Assuming cosine similarity then $\vec{q}_{opt}$ turns out to be the vector difference between the means of the relevant documents($C_R$) and that of the irrelevant documents($C_{NR}$). However since the given sets of relevant documents is just a subset of the complete set of relevant documents, we cannot use $D_R$ directly instead of $C_R$. Hence instead Rocchio proposed the following formulation of the feedback query vector $\vec{q}_f$:

$$\vec{q}_f = \alpha q_0 + \beta \frac{1}{|D_R|} \sum_{d_i \in D_R} \vec{d_i} - \gamma \frac{1}{|D_{NR}|} \sum_{d_i \in D_{NR}} \vec{d_i}$$

Hence starting from the initial query vector, $\vec{q}$ is moved in the direction of the centroid of $D_R$ while moving away from the centroid of $D_{NR}$. Note that since we are subtracting a term, we can have negative query weights. In such cases we set the weights of these terms to 0. In practice the $\beta > \gamma$ as positive feedback is generally more useful than negative feedback.

The Rocchio algorithm, apart from being one of the first algorithms for relevance feedback, is one of the first methods given for using negative data, i.e. irrelevant documents. Our method is thus related to the Rocchio algorithm as we also focus on using irrelevance data(which we learn ourselves) to improve relevance feedback. The difference lies in the fact that while Rocchio algorithm assumes that a set of relevant and a set of irrelevant documents are given, we make no such assumption and rather try to find these sets automatically.

### 3.4.2 Study of Negative Feedback

PRF techniques assume the top $k$ documents to be relevant, but that does not hold true in all cases. This is one of the main problems affecting PRF, as including irrelevant documents in the feedback set can affect performance. Recently there has been increased focus on negative feedback(i.e. Using Irrelevance Data) [WFZ08, WFZ07, ZHS09]. In negative feedback, you are not given positive data, but only negative data. Hence the standard methods of relevance feedback do not work. Wang et.al. studied different methods of using negative feedback. They discuss some of the negative feedback techniques from the domains of vector-space models, and from language models.

Zhang et.al were one of the first to attempt solving the problem of the feedback document set containing irrelevant documents. They treated the feedback set as a mixture of the relevant and an irrelevant distribution(where the distribution is over the set of terms). They proposed a distribution separation method to separate the irrelevant distribution from the feedback set mixture of relevant and irrelevant distributions, so as to obtain an approximation to the true relevance distribution. They made some assumptions which are critical for their method:

1. The feedback document set is a *linear combination* of an irrelevance distribution and a relevance distribution.
2. They assume that they are given "a small number (or a small portion, e.g., 10%) of top ranked irrelevant documents in the pseudo-relevance feedback set".
3. The given irrelevant distribution has a low correlation with the relevant distribution.

Some notations: Distribution of feedback documents is $M$; $R$ is the distribution of relevant documents in the feedback set; $I_G$ is the distribution of given irrelevant documents in the feedback set; $I_U$ is the distribution of unknown irrelevant documents in the feedback set. $l(A, B)$ stands for the linear combination of $A$ and $B$.

They first assume $M = l(l(R, I_U), I_G)$ with mixing parameter $\lambda$. Rewriting this they get $l(R, I_U) = \frac{1}{\lambda}M + (1 - \frac{1}{\lambda})I_G$. They thus estimate $\lambda_{opt}$ as the maximum value possible for which the estimated $l(R, I_U)$ has all positive values. Next they refine the values of $M, I_G$ and $\lambda_{opt}$ using a refinement parameter $\eta$. Finally they find a $\lambda$ for which the square of the correlation between $l(R, I_U)$ and $I_G$ is minimized, and thus use the estimated $l(R, I_U)$ as the approximate relevance distribution $R^*$. Thus in essence they try to extract a cleaner version of the relevant distribution, using the irrelevance data made available to them.

Note one of the key areas, where our work differs is that the Zhang et.al. assumed being given a set of irrelevant documents(they report results for considerably large fraction of 10%, 20% and 30%) from the feedback set. We feel this is not a feasible assumption to make in an automatic query expansion method. Our experiences have led us to realize the inherent complexity of the task of identifying irrelevant documents in the feedback set, and hence we feel making such an assumption is not practical for automatic expansion.

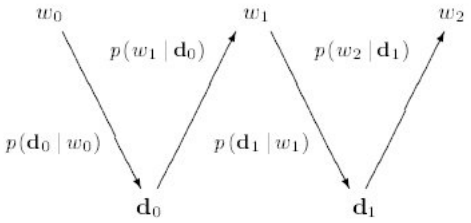


Figure 3.1: Random Walk from Words to Documents and Back

## 3.5 Random Walk Model by Lafferty-Zhai

Lafferty and Zhai in [LZ01] propose a random walk procedure, which alternates between documents and words in these documents. As seen in 3.1, a word $w$ goes to a document $d$ with probability given by $P(d|w)$, and a document goes to a word with probability $P(w|d)$. To get $P(w|d)$ the MLE distribution of terms in $d$, is smoothed by the collection model. The claim here is that this process, you would be able to obtain synonyms for words. However in practice, it has been observed that for a 1-step walk, the words are co-occurrence based, but as you take more steps you obtain synonyms. Since computation costs are heavy after the 2nd step, in practice only two steps are performed.

However the idea of using a random walk to obtain synonyms, is still a very interesting idea, There has been further extensions to this method as well [CTC05], but they have tended to be more complex, with only marginal gains. Inspired by this work and the work by [XFW02], we later propose a method to obtain a thesaurus, but by performing a random walk across words of different languages, rather than across words and documents.

# Chapter 4

# Multilingual Pseudo Relevance Feedback

In this chapter, we present a novel approach to PRF, which involves using one language (called the *assisting language*), to help improve PRF in the original language. We also present some of the experiments we performed, and the performance of the algorithm. In the next chapter we analyze the results we present in this chapter, and also analyze how the method depends on different factors, such as translation quality, source-assisting language similarities amongst others.

## 4.1 Motivation

PRF is a well-known method to perform automatic Query Expansion, which is known to improve retrieval performance. However it has two flaws: (1) *Lexical and Semantic Non-Inclusion*: The expansion terms found by PRF are primarily based on co-occurrence relationships, with the query terms. Hence other types of term associations such as morphological variants and synonyms are not explicitly captured. (2)*Instability and Lack of Robustness*: The assumption of the relevance of the top $k$ documents, does not hold in practice, resulting in *Topic Drift*. This causes the performance to decrease as well as a lack of robustness and sensitivity to initial retrieval performance.

We would like to come up with a framework which solves both of these problems together. One of the ideas used for making PRF more robust is using another, preferably larger collection, to obtain feedback terms as well. This is a popular approach used in the TREC tasks. The underlying concept here is that while the feedback set in one collection may have a lot of irrelevant documents, there is a good chance that this may not be the case in the other collection. Thus this approach can be thought of as a *Risk-Minimization* using two collections. Similarly, to solve the problem of synonyms and morphological variants being ignored, one method used is to utilize a lexical resource such as a dictionary or WordNet to obtain these terms. While each of these solve one of the two problems, they leave the other either untouched or worsened.

Another phenomenon which motivates our approach is the fact that retrieval is easier for some

| Symbol | Description |
|---|---|
| $\Theta_Q$ | Query Language Model |
| $\Theta^F_{L_1}$ | Feedback Language Model obtained from PRF in $L_1$ |
| $\Theta^F_{L_2}$ | Feedback Language Model obtained from PRF in $L_2$ |
| $\Theta^{Trans}_{L_1}$ | Feedback Model Translated from $L_2$ to $L_1$ |
| $\Theta^{Multi}_{L_1}$ | Final Feedback Model computed by MultiPRF |
| $P_{L_2 \rightarrow L_1}$ | Probabilistic Bi-Lingual Dictionary from $L_2$ to $L_1$ |
| $\beta, \gamma$ | Interpolation coefficients coefficients used in MultiPRF |

Table 4.1: Glossary of Mathematical Symbols used in explaining MultiPRF

languages due to availability of resources, but poor for other languages due to a lack of these resources. Particularly on the Web Scale, there is a huge disparity between languages when it comes to content. Hence we can use these content-rich/resource-rich languages to help retrieval in the content-poor/resource-poor languages; (the two phenomena are not independent, as most languages which are resource-poor also tend to lacking in content). Thus this motivates us to use English and other such languages to help improve retrieval in other languages. For example, English shares about 72% of the web content. Larger coverage typically ensures higher proportion of relevant documents in the top $k$ retrieval [HTH99]. This in turn ensures better PRF. Additionally, it is known that query processing in English is a simpler proposition than in most other languages due to English's simpler morphology and wider availability of NLP tools for English. Thus the approach we propose is especially attractive in the case of languages where the original retrieval is bad due to poor coverage of the collection and/or inherent complexity of query processing (for example *term conflation*) in those languages. For example, Hungarian has only 0.2% share of web content[1] with a rich morphology. Hence all these different factors, motivate us to use one language to help another.

## 4.2 MultiPRF Unraveled

In this section, we describe our approach: *Multilingual PRF*, in detail, and go into detail about each stage. The schematic of the MultiPRF approach is given in Figure 4.1.

Given a query $Q$ in the source language $L_1$, we automatically translate the query using a query translation system into the assisting language $L_2$. Using the translated query $Q_{L_2}$, we then rank the documents of a collection in language $L_2$, using the query likelihood ranking function [LZ03]. Using the top $k$ documents obtained via this ranking of the $L_2$ collection, we estimate the feedback model (in language $L_2$) using the MBF algorithm, which has been described earlier. Similarly, we also estimate a feedback model (in language $L_1$) using the original query and the top $k$ documents retrieved from the initial ranking on the $L_1$ collection (while again using the MBF algorithm to obtain the feedback model). Let the resultant feedback models, obtained via this process, be $\Theta^F_{L_2}$ and $\Theta^F_{L_1}$ respectively. For further clarification of notation, refer to Table 4.1, where all notation has been clearly explained.

[1] `http://www.netz-tipp.de/languages.html`

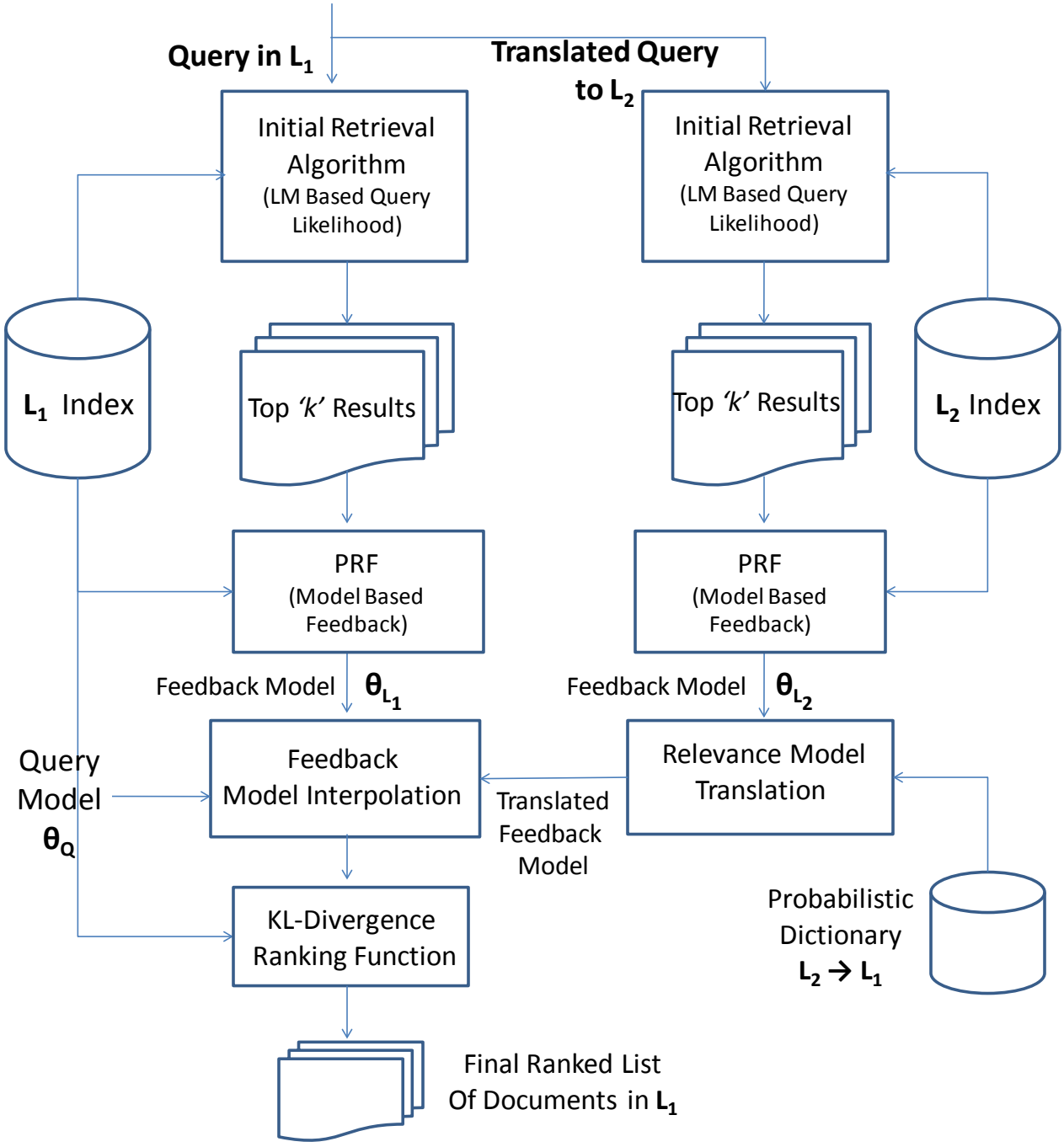


Figure 4.1: Schematic of the Multilingual Pseudo-Relevance Feedback Approach

| Source Term | Top Aligned Terms in Target |
|---|---|
| **French** | **English** |
| américain | american, us, united, state, america |
| nation | nation, un, united, state, country |
| étude | study, research, assess, investigate, survey |
| **German** | **English** |
| flugzeug | aircraft, plane, aeroplane, air, flight |
| spiele | play, game, stake, role, player |
| verhältnis | relationship, relate, balance, proportion |

Table 4.2: Top Translation Alternatives for some words in Probabilistic Bi-Lingual Dictionary

Next, the feedback model estimated in the assisting language, *i.e.* $\Theta^F_{L_2}$, is translated back into language $L_1$ using a probabilistic bi-lingual dictionary from $L_2 \rightarrow L_1$ as follows:

$$P(f|\Theta^{Trans}_{L_1}) = \sum_{\forall e \in \text{Vocab}(L_2)} P_{L_2 \rightarrow L_1}(f|e) \cdot P(e|\Theta^F_{L_2}) \tag{4.1}$$

where $P_{L_2 \rightarrow L_1}(f|e)$ refers to the probability of obtaining term $f$ of language $L_1$ from the term $e$ of language $L_2$. In our method, the probabilistic bi-lingual dictionary $P_{L_2 \rightarrow L_1}(f|e)$ used, was learned from a parallel sentence-aligned corpora in $L_1 - L_2$ based on word level alignments. Tiedemann [Tie01] has shown that the translation alternatives found using word alignments could be used to infer various morphological and semantic relations between terms. For example, in Table 4.2, we show the top English translation alternatives for some sample words from French and German, obtained from the *French→English* and *German→English* bilingual dictionaries respectively. For example, the French word *américain* (Meaning *american*) brings different variants of the translation like *american, america, us, united, state, america* which are lexically and semantically related. Hence, the probabilistic bi-lingual dictionary acts as a rich source of morphologically and semantically related feedback terms. Thus during the step for translating the feedback model $\Theta^F_{L_2}$ from $L_2$ back into $L_1$, given by Equation 4.1, related terms in $L_1$ are added to the translation model $\Theta^{Trans}_{L_1}$, due to source terms from feedback model $\Theta^F_{L_2}$.

The final MultiPRF model is obtained by interpolating the above translated feedback model with the original query model and the feedback model of language $L_1$ as given below:

$$\Theta^{Multi}_{L_1} = (1 - \beta - \gamma) \cdot \Theta_Q + \beta \cdot \Theta^F_{L_1} + \gamma \cdot \Theta^{Trans}_{L_1} \tag{4.2}$$

Since we want to retain the query focus during back translation the feedback model in $L_2$ is interpolated with the translated query before translation. The parameters $\beta$ and $\gamma$ control the relative importance of the original query model, feedback model of $L_1$ and the translated feedback model obtained from $L_1$ and are tuned based on the choice of collection in $L_1$ and $L_2$. Finally, the documents in the collection $L_1$ are ranked using the final feedback model $\Theta^{Multi}_{L_1}$.

| Language | CLEF Collection Identifier | Description | No. of Documents | No. of Unique Terms | CLEF Topics (No. of Topics) |
|---|---|---|---|---|---|
| **English** | EN-00+01+02 | LA Times 94 | 113005 | 174669 | - |
| | EN-03+05+06 | LA Times 94, Glasgow Herald 95 | 169477 | 234083 | - |
| | EN-02+03 | LA Times 94, Glasgow Herald 95 | 169477 | 234083 | 91-200 (67) |
| **French** | FR-00 | Le Monde 94 | 44013 | 127065 | 1-40 (29) |
| | FR-01+02 | Le Monde 94, French SDA 94 | 87191 | 159809 | 41-140 (88) |
| | FR-02+03 | Le Monde 94, French SDA 94-95 | 129806 | 182214 | 91-200 (67) |
| | FR-03+05 | Le Monde 94, French SDA 94-95 | 129806 | 182214 | 141-200,251-300 (99) |
| | FR-06 | Le Monde 94-95, French SDA 94-95 | 177452 | 231429 | 301-350 (48) |
| **German** | DE-00 | Frankfurter Rundschau 94, Der Spiegel 94-95 | 153694 | 791093 | 1-40 (33) |
| | DE-01+02 | Frankfurter Rundschau 94, Der Spiegel 94-95, German SDA 94 | 225371 | 782304 | 41-140 (85) |
| | DE-02+03 | Frankfurter Rundschau 94, Der Spiegel 94-95, German SDA 94-95 | 294809 | 867072 | 91-200 (67) |
| | DE-03 | Frankfurter Rundschau 94, Der Spiegel 94-95, German SDA 94-95 | 294809 | 867072 | 141-200 (51) |
| **Finnish** | FI-02+03+04 | Aamulehti 94-95 | 55344 | 531160 | 91-250 (119) |
| | FI-02+03 | Aamulehti 94-95 | 55344 | 531160 | 91-200 (67) |
| **Dutch** | NL-02+03 | NRC Handelsblad 94-95, Algemeen Dagblad 94-95 | 190604 | 575582 | 91-200 (67) |
| **Spanish** | ES-02+03 | EFE 94, EFE 95 | 454045 | 340250 | 91-200 (67) |
| **Hungarian** | HU-05 | Magyr Hirlap 2002 | 49530 | 256154 | 251-300 (48) |

Table 4.3: Details of the CLEF Datasets used for Evaluating the MultiPRF approach. The number shown in brackets of the final column CLEF Topics indicate the actual number of topics used during evaluation.

## 4.3 Experimental Setup

We evaluate the performance of our system using the standard CLEF evaluation data in seven languages, widely varying in their familial relationships - Dutch, German, English, French, Spanish, Finnish and Hungarian using more than 600 topics. To begin, we report results of experiments with English as the assisting language. We later relax this, and report results for all possible source-assisting language pairs. The details of the collections, their corresponding topics and the assisting collections used for MultiPRF are given in Table 4.3. Note that we choose the English assisting collection such that the coverage of topics is similar to that of the original corpus so as to get meaningful feedback terms. Hence this is a slightly weaker requirement than *comparable corpora*. In all our experiments, as queries, we only use the *title* field of the topics. We ignore the topics which have no relevant documents as the true performance on those topics cannot be evaluated, and hence tp prevent the obfuscation of results we exclude them.

We use the Terrier IR platform [OAP+06] to perform all our experiments, due to its' support of European Languages. For indexing the documents in the collection, we use the inbuilt system of Terrier. We also perform standard tokenization, stop word removal and stemming in all languages. We use the Porter Stemmer for English and the stemmers available through the Snowball[2] package for Spanish, French, German, Dutch, Finnish and Hungarian. Other than these, we do not perform any other processing on languages, other than French. However, in the case of French, since some function words like *l'*, *d' etc.,* occur as prefixes to a word, we strip them off during indexing and

[2] `http://snowball.tartarus.org/index.php`

query processing. We do this to prevent the baseline performance to decrease, and thus get a more realistic idea of the performance of the method in French. Also note that due to the lack of a publicly available decompounder for German, Dutch, Finnish and Hungarian, the performance of MultiPRF and the baselines on these languages, is affected. We use standard evaluation measures like *MAP*, *P@5* and *P@10* for evaluation. Additionally, for assessing robustness, we use the Geometric Mean Average Precision (GMAP) metric [Rob06] which is also used in the TREC Robust Track

The probabilistic bi-lingual dictionary used in MultiPRF, and all related experiments, was learnt automatically by running GIZA++: a word alignment tool [ON03] on a parallel sentence aligned corpora. For all the above language pairs except Hungarian, we used the *Europarl Corpus* [Phi05] and in case of Hungarian-English, we used the Hunglish Corpus [3]. Note that, since in IR we only deal with stemmed terms, the bilingual dictionaries also contain only stemmed terms. Hence we first tokenized and did all the required pre-processing, as mentioned in the above paragraph, on the Europarl corpora, and then stemmed it, before we ran GIZA++.

Since MultiPRF requires a query translation module for all language pairs, due to its' ease of availability we used Google Translate[4] as the query translation system. Google Translate has also been shown to perform well for the task [WHJG08]. Later, we show that our approach is not dependent on Google Translate, and report results using a basic SMT system for query translation. We also evaluate the quality of the above Query Translation systems and analyze their impact on the quality of our results. In the pathological case of term not being found in the assisting language after query translation, we only perform MBF on the source language $L_1$. Note that, Google Translate too is not always accurate, and in recent research, most query translation system used for CLIR are shown to be about 95% accurate, which as we will see later, is better than the performance of Google translate.

We use the MBF approach, as explained in previous chapters, as a baseline for comparison. We use two-stage Dirichlet smoothing with the optimal parameters tuned based on the collection [ZL04]. We tune the parameters of MBF, specifically $\lambda$ and $\alpha$, and choose the values which give the optimal performance on a given collection. We uniformly choose the top $k$ documents as the feedback set, with $k$ set as 10. We observe that the optimal values of both interpolation coefficients $\beta, \gamma$ in MultiPRF are almost uniform across collections and vary in the range of [0.40,0.48].

## 4.4 Results

We initially test the performance of our approach, using English as the *assisting language* (Since English has resources and content, and thus a good assisting language). As mentioned above, Google Translate is used as the Query Translation System. The results for the approach are given in Table 4.4. As can be seen, the results show that the MultiPRF approach significantly outperforms the MBF approach across all datasets of all the chosen languages. We consistently observe significant improvements in all metrics, namely MAP (between 4% to 8%), P@5 (between 4% to 39%) and P@10 (around 4% to 22%). The MultiPRF approach is also more robust than plain MBF as reflected in the improvements obtained in GMAP scores (between 15% to 730%). This

[3] `http://mokk.bme.hu/resources/hunglishcorpus`

[4] `http://translate.google.com`

| | MAP | | | P@5 | | | P@10 | | | GMAP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Collection** | **MBF** | **MultiPRF** | **% Improv.** | **MBF** | **MultiPRF** | **% Improv.** | **MBF** | **MultiPRF** | **% Improv.** | **MBF** | **MultiPRF** | **% Improv.** |
| **FR-00** | 0.4220 | 0.4393 | 4.10 | 0.4690 | 0.5241 | **11.76**‡ | 0.4000 | 0.4000 | 0.00 | 0.2961 | 0.3413 | **15.27** |
| **FR-01+02** | 0.4342 | 0.4535 | **4.43**‡ | 0.4636 | 0.4818 | 3.92 | 0.4068 | 0.4386 | **7.82**‡ | 0.2395 | 0.2721 | **13.61** |
| **FR-03+05** | 0.3529 | 0.3694 | **4.67**‡ | 0.4545 | 0.4768 | **4.89**‡ | 0.4040 | 0.4202 | **4**‡ | 0.1324 | 0.1411 | 6.57 |
| **FR-06** | 0.3837 | 0.4104 | 6.97 | 0.4917 | 0.5083 | 3.39 | 0.4625 | 0.4729 | 2.25 | 0.2174 | 0.2810 | **29.25** |
| **DE-00** | 0.2158 | 0.2273 | 5.31 | 0.2303 | 0.3212 | **39.47**‡ | 0.2394 | 0.2939 | **22.78**‡ | 0.0023 | 0.0191 | **730.43** |
| **DE-01+02** | 0.4229 | 0.4576 | **8.2**‡ | 0.5341 | 0.6000 | **12.34**‡ | 0.4864 | 0.5318 | **9.35**‡ | 0.1765 | 0.2721 | 9.19 |
| **DE-03** | 0.4274 | 0.4355 | 1.91 | 0.5098 | 0.5412 | 6.15 | 0.4784 | 0.4980 | 4.10 | 0.1243 | 0.1771 | **42.48** |
| **FI-02+03+04** | 0.3966 | 0.4246 | **7.06**‡ | 0.3782 | 0.4034 | **6.67**‡ | 0.3059 | 0.3319 | **8.52**‡ | 0.1344 | 0.2272 | **69.05** |
| **HU-05** | 0.3066 | 0.3269 | **6.61**‡ | 0.3542 | 0.4167 | **17.65**‡ | 0.3083 | 0.3292 | **6.76**‡ | 0.1326 | 0.1643 | **23.91** |

Table 4.4: Results comparing the performance of MultiPRF approach over the baseline MBF approach on CLEF collections. Results marked as ‡ indicate that the improvement was found to be statistically significant over the baseline at 90% confidence level ($\alpha = 0.01$) when tested using a paired two-tailed t-test.

could be attributed in part to the reduced sensitivity of our approach to the number of relevant documents in the feedback set of the source language. In the next chapter, on analyzing the overall results, we see that MultiPRF leverages the performance of the assisting language and adds relevant morphological variants and synonyms in addition to co-occurrence based term relations. Besides this, it also improves the performance of some queries where the PRF performance was poor to start with, by bringing in related terms through PRF in $L_2$ and back translation.

As shown in the Table, most of the gains achieved in MAP, P@5 and P@10 are statistically significant. Note that, we cannot do significance tests on GMAP. Thus we tend to see the biggest improvements in those languages, which have poor performance to begin with. Hence, as per our intuition, MultiPRF provides a risk-minimization framework, thus improving robustness. We also noted that the number of failed queries (*i.e.* queries with MAP value $\leq 0.1$) decreases as compared to the baseline.

We next remove the restriction of English being the assisting language, and instead compare all source-assisting language pairs. However, we no longer consider Hungarian, since it does not have a parallel corpus with the other languages, and hence no bilingual dictionary. We will especially focus our attention towards languages which are closely related or in the same family. Amongst the languages, we have chosen the different linguistic families which find representation are:

- *Germanic Family*: This family includes *English, German and Dutch*
- *Romance Language Family*: This set of languages includes *French and Spanish*
- *Fino-Ugric Family*: *Finnish* belongs to this family.

| Language | Properties |
|---|---|
| English | Very simple Morphology |
| French | Complex Morphology |
| Spanish | Complex Morphology |
| Dutch | Complex Morphology + Word Compounding |
| German | Complex Morphology + Severe Word Compounding |
| Finnish | Complex Morphology + Word Compounding +Agglutination |

Table 4.5: Linguistic Characteristics of the different Languages

| MBF Performance | MAP | P@5 | P@10 | GMAP |
|---|---|---|---|---|
| **English** | 0.4495 | 0.4955 | 0.4328 | 0.2574 |
| **German** | 0.4033 | 0.5134 | 0.4746 | 0.191 |
| **Dutch** | 0.4153 | 0.5045 | 0.4657 | 0.122 |
| **Spanish** | 0.4805 | 0.6388 | 0.5731 | 0.3015 |
| **French** | 0.4356 | 0.4776 | 0.4194 | 0.2507 |
| **Finnish** | 0.3578 | 0.3821 | 0.3105 | 0.1604 |

Table 4.6: Baseline MBF Performance across the 02-03 collections of all languages

However, we also note that English and French are related to each other, despite not being from the same family. These two languages share common words in their vocabularies. We will also keep in mind, the different linguistic challenges associated with each of these languages. This can be seen in Table 4.5.

For the above six languages, we indexed the CLEF collections of the years 2002 and 2003. In order to ensure that the results are comparable across languages, we selected topics from the years 2002, 2003 (from CLEF Topics 91-200) that have relevant documents in all the languages. The number of such common topics was 67. For each source language, we use the other languages as assisting collections and study the performance of MultiPRF. Since query translation quality varies across language pairs, in order to eliminate its effect, we skip the query translation step and use the corresponding original topics for each target language instead. The baseline MBF performances on these collections is provided in Table 4.6.

| Source \ Assisting | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 0.4513 | 0.4475 | 0.4695 | 0.4665 | 0.4416 |
| **German** | 0.4427 | - | 0.4306 | 0.4404 | 0.4104 | 0.3993 |
| **Dutch** | 0.4361 | 0.4344 | - | 0.4227 | 0.4304 | 0.4134 |
| **Spanish** | 0.4665 | 0.4773 | 0.4733 | - | 0.4839 | 0.4412 |
| **French** | 0.4591 | 0.4514 | 0.4409 | 0.4712 | - | 0.4354 |
| **Finnish** | 0.3733 | 0.3559 | 0.3676 | 0.3594 | 0.371 | - |

Table 4.7: MAP Values of MultiPRF for all different source-assisting language pairs

| Source \ Assisting | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 0.5104 | 0.5104 | 0.5343 | 0.5403 | 0.4806 |
| **German** | 0.606 | - | 0.5672 | 0.594 | 0.5761 | 0.5552 |
| **Dutch** | 0.5761 | 0.5552 | - | 0.5403 | 0.5463 | 0.5433 |
| **Spanish** | 0.6507 | 0.6448 | 0.6507 | - | 0.6478 | 0.597 |
| **French** | 0.4925 | 0.4776 | 0.4776 | 0.4995 | - | 0.4955 |
| **Finnish** | 0.4149 | 0.385 | 0.388 | 0.388 | 0.3911 | - |

Table 4.8: P@5 Values of MultiPRF for all different source-assisting language pairs

| Source \ Assisting | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 0.4373 | 0.4358 | 0.4597 | 0.4582 | 0.4164 |
| **German** | 0.5373 | - | 0.503 | 0.5299 | 0.494 | 0.5 |
| **Dutch** | 0.5254 | 0.497 | - | 0.4776 | 0.5134 | 0.4925 |
| **Spanish** | 0.5791 | 0.5791 | 0.5761 | - | 0.5866 | 0.5567 |
| **French** | 0.4463 | 0.4313 | 0.4373 | 0.4448 | - | 0.4209 |
| **Finnish** | 0.3567 | 0.31 | 0.3253 | 0.32 | 0.3239 | - |

Table 4.9: P@10 Values of MultiPRF for all different source-assisting language pairs

| Source \ Assisting | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 0.2912 | 0.2557 | 0.2573 | 0.2789 | 0.2128 |
| **German** | 0.273 | - | 0.2179 | 0.2321 | 0.237 | 0.1641 |
| **Dutch** | 0.2091 | 0.1836 | - | 0.1434 | 0.1858 | 0.1441 |
| **Spanish** | 0.3229 | 0.3235 | 0.2913 | - | 0.3377 | 0.2697 |
| **French** | 0.2768 | 0.2825 | 0.2665 | 0.2624 | - | 0.2041 |
| **Finnish** | 0.1915 | 0.1587 | 0.1762 | 0.1658 | 0.1645 | - |

Table 4.10: GMAP Values of MultiPRF for all different source-assisting language pairs

| Source \ Assisting → | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 0.400445 | -0.44494 | **4.449388** | **3.78198** | -1.75751 |
| **German** | **9.769402** | - | **6.769154** | **9.199107** | 1.760476 | -0.99182 |
| **Dutch** | **5.008428** | **4.599085** | - | 1.781844 | **3.635926** | -0.4575 |
| **Spanish** | -2.91363 | -0.66597 | -1.49844 | - | 0.707596 | -8.17898 |
| **French** | **5.394858** | **3.627181** | 1.216713 | **8.172635** | - | -0.04591 |
| **Finnish** | **4.332029** | -0.53102 | 2.73896 | 0.447177 | **3.689212** | - |

Table 4.11: Percentage Increase Values of MAP of MultiPRF over MBF

| Source \ Assisting → | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| **English** | - | 7.61 | 10.51 | *88.14* | 62.53 | 58.34 |
| **German** | **93.34** | - | *86.15* | **99.09** | 38.53 | 21.64 |
| **Dutch** | *85.79* | **92.28** | - | 53 | 76 | 10.27 |
| **Spanish** | 55.77 | 25.76 | 44.65 | - | 19.15 | **96.72** |
| **French** | *89.24* | *83.71* | 31.34 | **99.52** | - | 0.69 |
| **Finnish** | 52.82 | 15.83 | *81.03* | 14.04 | 77.65 | - |

Table 4.12: Significance Values for 2-tailed T-Test for MAP performance of MultiPRF vs MBF

In Table 4.7, the MAP value performance of the MultiPRF approach is given for different source-assisting language pairs. Similarly in Table 4.8, Table 4.9 and Table 4.10 we have the P@5, P@10 and GMAP performances respectively, of all these different pairs of languages. We also have the percentage improvements of MAP values of MultiPRF over MBF for these different pairs in Table **??**, and the corresponding significance values in Table 4.12.

Hence as we can see, except for a few cases MultiPRF gives significant and consistent improvement over baseline values. These improvements are not only in MAP, but also in P@5 and P@10. Furthermore large improvements in GMAP values also indicate the method to be more robust as well. At the same time, we also note that MultiPRF does not give improvements in all cases. As we see in Table 4.7, when Finnish is used as the assisting language, there is either no change or decrease in performance with respect to the baselines. This indicates that the more complex a language and its' characteristics, the less likely it is to improve performance. We also see that in the case of Spanish as a source language, we do not get any improvements in the MAP values, except with French as the assisting language, because the baseline performance is very strong. Thus since the initial MAP performance is strong and robust, the potential for MultiPRF to improve this is reduced, thus resulting in MultiPRF performing very similar to the baselines. However note that the P@5, P@10 and GMAP values improve in MultiPRF as compared to the baselines for almost all assisting languages, thus highlighting the benefits of MultiPRF even in these settings and showing it to be more robust. In the next chapter we get into more analysis of the method and discuss various aspects of it.

# Chapter 5

# Analyzing the Results of MultiPRF

In the previous chapter we described the MultiPRF approach. We also found MultiPRF to give consistent and significant improvements over the baselines across all metrics, namely MAP, P@5, P@10 and GMAP. In this chapter, we take a closer look at the results and analyze them. We do a query-level analysis, to ascertain the reasons MultiPRF improves over PRF. We also analyze the sensitivity of the algorithm to the different components of the system. We also try looking into the performance dependence of the algorithm on the similarity between the source and assisting languages.

## 5.1 Query-Level Analysis

To get a deeper understanding for the working of the method and to illustrate the qualitative improvement in feedback terms, a detailed analysis of a few representative queries is presented in Table 5.1. These queries are from the French and German collections, with English used as the assisting language, and Google Translate as the query-translation system. While the first four queries, represent MultiPRF improving over baseline MBF; the last two queries are examples of where the baseline MBF performance was very competitive, thus causing MultiPRF to actually performing worse then MBF. After studying many queries and understanding the reasons for improvements obtained by MultiPRF approach, we postulate that these improvements could be mainly attributed to one of the following three reasons:-

1. Retrieval performance in $L_2$ is good and the resultant feedback model contains a lot of relevant terms, which when brought back to $L_1$ via back-translation leads to improvement.
2. During the back-translation process, important synonyms and popular morphological variants (inflectional forms) of key terms are found, which otherwise were missing from the MBF model.
3. A combination of both the above factors.

| TOPIC NO. | ORIGINAL QUERY | TRANSLATED ENGLISH QUERY | MBF MAP | MPRF MAP | MBF - Top Representative Terms (With meaning) | MultiPRF - Top Representative Terms (With meaning) |
|---|---|---|---|---|---|---|
| FRENCH '00. TOPIC 33 | Tumeurs et génétique | Tumors and Genetics | 0.0414 | 0.2722 | malad (ill), tumeur (tumor), recherch (research), canc (cancer), yokoham, hussein | tumor (tumor), génet, canc, gen, malad, cellul (cellular), recherch |
| FRENCH '03. TOPIC 198 | Oscar honorifique pour des réalisateurs italiens | Honorary Oscar for Italian filmmakers | 0.1238 | 0.4324 | italien, président (president), oscar , gouvern (governer) , scalfaro , spadolin | film, italien, oscar, honorair (honorary) , cinem (film), cinéast (filmmaker), réalis (achieve), produit(product) |
| FRENCH '06. TOPIC 317 | Les Drogues Anti-cancer | The Anti-Cancer Drugs | 0.001 | 0.1286 | drogu (drugs), anti , trafic (trafficking), entre (between), légalis (legalise), canc , malad , cocaïn , afghanistan , iran | canc, drogu, recherch, malad, trait, taxol, glaxo, cancer |
| GERMAN '02. TOPIC 115 | Scheidungsstatistiken | Divorce Statistics | 0.2206 | 0.4463 | prozent (percent), unterstutz (supporters), frau (woman), minderjahr (underage), scheidung (divorce) | statist, scheidung , zahl (number), elt (parent), kind (child), famili, geschied (divorced), getrennt (separated), ehescheid (divorce) |
| GERMAN '03. TOPIC 147 | Ölunfälle und Vögel | Birds and Oil Spills | 0.0128 | 0.1184 | rhein (rhine), olunfall (oil spill), fluss (river), ol (oil) , heizol (fuel/oil), tank (tanker) | ol, olverschmutz (oil pollution), vogel (bird), erdol (petroleum), olp (oil slick), olunfall , gallon, vogelart (bird species) |
| FRENCH '05. TOPIC 274 | Bombes actives de la Seconde Guerre Mondiale | Active bombs of the Second World War | 0.6182 | 0.3206 | bomb, guerr(war), mondial (world), vill(city), découvert(discovery), second, explos, alemagn (germany), allemand (german) | guerr, mond(world), deuxiem (second), activ, bombard, japon(japan), hiroshim, nagasak, atom, nucléair (nuclear) |
| GERMAN '03. TOPIC 188 | Deutsche Rechtschreibreform | German spelling reform | 0.8278 | 0.6776 | deutsch, reform, spiegel (reflect), rechtschreibreform (spelling reform), sprach (language), osterreich (austria), rechtschreib (spelling), wien (vienna), schweiz (switzerland) | deutsch, reform, deutschland (Germany), clinton, deutlich (clearly), president , berlin, europa, gipfel(summit), bedeut(important) |

Table 5.1: Qualitative comparison of feedback terms given by MultiPRF and MBF on representative queries where positive and negative improvements were observed in French and German collections with English as assisting language.

Below are the detailed explanation of some sample queries, which demonstrate the robustness of MultiPRF and the reduced sensitivity to the relevance of the top documents from the initial retrieval.

- For example, consider the French Query *"Oscar honorifique pour des réalisateurs italiens"*, meaning "Honorary Oscar for Italian Filmmakers". Model-Based Feedback on French expands the query using the top retrieved documents of the initial retrieval. However, here it introduces significant topic drift towards Oscar Scalfaro (a former Italian President) and Italian politics thus causing words such as {*scalfaro, spadolin, gouvern*}. However, feedback in English produces relevant terms, which on translation back into French, introduces terms such as {*cinem, cinéast, réalis*}. This wrenches back the focus of the query from the political domain to the intended film domain, thus leading to performance increase.
- Another example of the above phenomenon is the query *"Les Drogues Anti-Cancer"* meaning Anti-Cancer Drugs. Here too MBF causes drift away from the intended meaning and instead to Drug-Trafficking, by introducing terms such as {*traffic, entre, afghanistan*}, which causes very poor performance on the query. MultiPRF however utilizes the good feedback performance of English on this query, to generate a set of very relevant French terms such as {*recerch, taxol, glaxo*}. Hence the drift from the intended meaning towards drug-trafficking is corrected.

Apart from this we also see improvements on queries due to introduction of synonyms and other semantically related terms. For example on the German query *"Ölunfälle und Vögel"* mean-

| Corpus | Google Translate | SMT |
|---|---|---|
| FR-01+02 | 0.93 | 0.67 |
| FR-03+05 | 0.88 | 0.77 |
| DE-01+02 | 0.93 | 0.64 |
| DE-03 | 0.81 | 0.58 |

Table 5.2: Comparison of Query Translation Quality using Google Translate and SMT system trained on Europarl Corpus on a scale of 0-1.

ing "Birds and Oil Spills", MBF performs poorly with many irrelevant terms introduced in the feedback model. However English finds some relevant terms, and additionally adds many terms to the feedback model, which are synonyms/semantically related to oil spills and birds, such as {*olverschmutz,ol,olp,vogelart*}. This helps in bringing up more relevant documents while reducing drift.

## 5.2 Effect of Query Translation Quality

As seen in the system architecture of MultiPRF, there is a critical component in the form of a *Query Translation Module*. As can be envisaged, accurate Query Translation is fundamental to MultiPRF. As explained earlier, we chose *Google Translate* mainly due to its ease of availability. We also note that Google Translate is not ideal, and there have been recent methods proposed to obtain very high accuracy levels for query translation for CLIR tasks. In this section, we study the impact of varying translation quality on the performance of our approach. We would like our approach to be robust, to an extent, from suboptimal quality of query translation. Hence we experiment by using such a query translation system. To this purpose, we trained a Statistical Machine Translation (SMT) system, on the French-English and German-English language pairs, by running Moses [HBCb$^+$07] on the Europarl corpora. The SMT system, we created in this manner, was quite simple as we did not perform any language-specific processing or any parameter tuning to improve the performance of the system. Another limitation of the system created in thus manner is that it is limited by the domain of the parallel corpora, which in the case of the Europarl Corpora is *parliamentary proceedings*.

To quantitatively judge the quality of these different translation systems, we used human annotations to score these different systems. Thus, to correlate the translation quality with the performance of MultiPRF, we evaluated the query translations produced by Google Translate and SMT system on a three-point scale between 0 and 1 (0 - Completely Wrong Translation, 0.5 - Translation not optimal but query intent partially conveyed and 1 - Query intent completely conveyed). The results are shown in Table 5.2. We compare the performance of MultiPRF using Google Translate, Basic SMT system, and Ideal query translations. The ideal translations were obtained by manually fixing some of the errors in the above two systems. Again the ideal translations are not truly ideal, as the errors fixed were by non-native speakers of these languages. However their quality was very close to being optimal. Thus, the performance reported on ideal query translations gives an idea of the upper bound on the performance of MultiPRF. The results of our evaluation are shown in Table 5.3.

| | MAP | | | | P@5 | | | | P@10 | | | | GMAP | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MBF | MPRF SMT | MPRF GT | MPRF Ideal | MBF | MPRF SMT | MPRF GT | MPRF Ideal | MBF | MPRF SMT | MPRF GT | MPRF Ideal | MBF | MPRF SMT | MPRF GT | MPRF Ideal |
| FR-01+02 | 0.4342 | 0.4494 | 0.4535 | 0.4633 | 0.4636 | 0.4818 | 0.4818 | 0.4864 | 0.4068 | 0.4239 | 0.4386 | 0.4477 | 0.2395 | 0.245 | 0.2721 | 0.2965 |
| FR-03+05 | 0.3529 | 0.3576 | 0.3694 | 0.3762 | 0.4545 | 0.4707 | 0.4768 | 0.4889 | 0.404 | 0.4141 | 0.4202 | 0.4323 | 0.1324 | 0.1329 | 0.1411 | 0.1636 |
| DE-01+02 | 0.4229 | 0.4275 | 0.4576 | 0.4639 | 0.5341 | 0.5523 | 0.6 | 0.6 | 0.4864 | 0.5125 | 0.5271 | 0.5386 | 0.2492 | 0.2032 | 0.2721 | 0.2816 |
| DE-03 | 0.4274 | 0.4236 | 0.4355 | 0.4388 | 0.5098 | 0.5294 | 0.5412 | 0.5451 | 0.4784 | 0.4863 | 0.498 | 0.4922 | 0.1243 | 0.1225 | 0.1771 | 0.1981 |

Table 5.3: Results comparing the performance of MultiPRF approach (with English as assisting Language) over the baseline MBF approach with Google Translate and another SMT system trained using Europa corpus.

| | Assisting Language | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Source Language** | **English** | **German** | **Dutch** | **Spanish** | **French** | **Finnish** | **Source Monolingual MBF** | **Assisting Language Rank** |
| **English** | - | **0.4513** | 0.4476 | **0.4695** | **0.4665** | 0.4416 | 0.4495 | |
| **German** | **0.4427** | - | **0.4306** | **0.4404** | 0.4104 | 0.3993 | 0.4033 | **Rank 1** |
| **Dutch** | **0.4361** | **0.4355** | - | 0.4227 | **0.4304** | 0.4134 | 0.4153 | **Rank 2** |
| **Spanish** | 0.4665 | **0.4773** | **0.4733** | - | **0.4839** | 0.4412 | 0.4805 | **Rank 3** |
| **French** | **0.4591** | **0.4514** | 0.4409 | **0.4712** | - | 0.4354 | 0.4356 | |
| **Finnish** | **0.3733** | 0.3559 | **0.3676** | 0.3594 | **0.371** | - | 0.3578 | |

Table 5.4: Results showing the MAP performance of MultiPRF with different source and assisting languages. The rank of assisting languages with respect to performance on a given source language is highlighted using different color codes. The intra familial affinity could be observed from the elements close to the diagonal.

As expected, the performance of MultiPRF on ideal translations is the best followed by Google Translate and the Basic SMT system. The results demonstrate that translation using the basic SMT system improves over monolingual MBF, especially P@5 and P@10. This shows that the performance of MultiPRF improves with any reasonably good query translation system.

## 5.3 Effect of Assisting Language Choice on MultiPRF Accuracy

As seen in the last chapter, MultiPRF continues to perform well, even when the assisting language is not English. In this section, we try to understand if the assisting language plays a part in the performance of the system. We also investigate if the relation between the source and assisting language pairs affects performance, and if so, how. To this effect, the results from the previous section are repeated in Table 5.4, but with the top 3 ranked assisting language, for each source language marked.

| Source \ Assisting | English | German | Dutch | Spanish | French | Finnish |
|---|---|---|---|---|---|---|
| English | - | 3 | 4 | 1 | 2 | 5 |
| German | 1 | - | 3 | 2 | 4 | 5 |
| Dutch | 1 | 2 | - | 4 | 3 | 5 |
| Spanish | 4 | 2 | 3 | - | 1 | 5 |
| French | 2 | 3 | 4 | 1 | - | 5 |
| Finnish | 1 | 5 | 3 | 2 | 4 | - |
| Avg. Posn. as Assisting | 1.80 | 3.00 | 3.40 | 2.40 | 2.40 | 5.00 |

Table 5.5: Performance of Different Languages as Assisting Languages

To investigate if some languages *assist* more than other languages, as can be expected, we rank the performance of the possible assisting languages, for a fixed source language. The results for the same can be found in Table 5.5.

From the results, we observe the following:

- English is a very good assisting language, as expected. It is the best assisting language in 3 out of the 5 cases, and comes $2^{nd}$ in one. The case of Spanish source-English assisting, is an outlier to this theory. One possible reason for this is the poor quality of the English to Spanish multilingual dictionary.
- Finnish is a very poor assisting language. This again is expected, since Finnish is linguistically the most complex of the chosen languages, and has the worse monolingual performance as well. In all the cases, it ranks as the worst assisting language, and using it in MultiPRF leads to zero or negative improvements.
- Many pairs of languages, seem to do well with each other. These are the pairs highlighted in Table 5.5.

Thus we observe that besides English, other languages such as French, Spanish, German and Dutch can also act as good assisting languages and help in improving performance over monolingual MBF. We also observe that the best assisting language varies with the source language. However, the crucial factors of the assisting language which influence the performance of MultiPRF are:

| Language | Source Collection | Assisting Collection | No. of Docs. in Source Collection | No. of Docs. in Assisting Collection | MAP | GMAP |
|---|---|---|---|---|---|---|
| **German** | DE-01+02 | DE-03 | 225371 | 294809 | 0.4445 | 0.2328 |
| | DE-01+02 | EN-00+01+02 | 225371 | 113005 | **0.4576** | **0.2721** |
| **French** | FR-01+02 | FR-06 | 87191 | 177452 | 0.4394 | 0.2507 |
| | FR-01+02 | EN-00+01+02 | 87191 | 113005 | **0.4535** | **0.2721** |

Table 5.6: Comparison of MultiPRF performance with MBF using an assisting collection in the same language. The coverage of the source and assisting collections is also given for comparison.

- *Monolingual PRF Performance*: The main motivation for using a different language was to get good feedback terms, especially in case of queries which fail in the source language. Hence, an assisting language in which the monolingual feedback performance itself is poor, is unlikely to give any performance gains. This observation is evident in case of Finnish, which has the lowest Monolingual MBF performance. The results show that Finnish is the least helpful of assisting languages, with performance similar to those of the baselines. We also observe that the three best performing assistant languages, i.e. English, French and Spanish, have the highest monolingual performances as well, thus further validating the claim.
- *Familial Similarity Between Languages*: We observe that the performance of MultiPRF is good if the assisting language is from the same language family. For example, in the Germanic family, the source-assisting language pairs English-German, German-English, Dutch-English, Dutch-German and German-Dutch show good performance. Similarly, in Romance family, the performance of Spanish-French and French-Spanish confirms this behaviour. Thus similarity in languages leads to better performance. This can again be understood by considering the quality of the bilingual dictionaries, in the case of these languages. Since these languages are from the same family and related to each other, we can firstly expect them to share common sections of vocabulary. We can also expect the overall quality of the dictionaries for these pairs to be better than for some of the other pairs.
- *Processing Complexity of the languages*: This is correlated to the first point also. As we see, English is the nest assisting language as well as the easiest to process. Finnish, which is the most difficult to process, uniformly ranks as the worst assisting language. This affects performance in the form of the quality of the Bilingual dictionary learnt. We observed that the quality of dictionaries, which are complex to process was suboptimal.

In some cases, we observe that MultiPRF scores decent improvements even when the assisting language does not belong to the same language family as witnessed in French-English and English-French. This is primarily due to their strong monolingual MBF performance.

| Collection | MAP | P@5 | P@10 | GMAP |
|---|---|---|---|---|
| **FR-01+02** | 0.4394 | 0.4682 | 0.4341 | 0.2507 |
| **DE-01+02** | 0.4344 | 0.5409 | 0.4966 | 0.2328 |

Table 5.7: Performance of MultiPRF using an assisting collection in the same language. The coverage of the source and assisting collections is also given for comparison.

## 5.4 Comparison with Assisting Collection in Same Language

One of the prime reasons for improvement in MultiPRF performance is good monolingual performance of assisting collection. The natural question which may then arise is whether the assisting collection needs to be in a different language. In this section, we study the performance of MultiPRF when the assisting collection is in the same language. Given a query, we use MBF on both source and assisting collections and interpolate the resultant feedback models. The final interpolated model is used to rerank the corpus and produce the final results. For the experiments, we use the French and German collections (FR-01+02, DE-01+02) since they have additional collections (FR-06, DE-03) with larger coverage in their own language. The results of comparison are shown in Table 5.6. The values of other measures such as P@5 and P@10 are given in Table 5.7.

From the results, we notice that although the coverage of assisting collections in the source language is more than that of English, MBF still performs poorly when compared to MultiPRF. This can be attributed to the following reasons a) the MBF performance of a query, which is ambiguous or hard in the source language collection, will be bad due to the poor quality of top $k$ documents retrieved during initial retrieval. The quality of the top $k$ documents will not change if the same ambiguous query is given to assisting collection in the source language. However, if source and assisting languages differ, the ambiguity may get resolved during translation causing an improvement in MBF performance. The above intuition is confirmed by the decrease in robustness, as reflected in the GMAP scores, when the source and target languages are same. b) it still suffers from the fundamental limitation of monolingual PRF *i.e.* the expansion terms included are only based on co-occurrence relations and does not include lexically and semantically related terms.

## 5.5 Comparison with Thesaurus Based Expansion in Source Language

As discussed earlier, another major source of improvement in MultiPRF is due to the inclusion of lexically and semantically related terms. However, this alone does not justify the use of an assisting collection in a different language since the same effect could be achieved by using *thesaurus based expansion* in the source language. In this section, we show that augmenting MBF with both thesaurus based expansion and assisting collection in the same language is not effective when compared to MultiPRF.

Since there is no publicly available thesauri for the above mentioned European languages, as proposed in Xu *et al.* [XFW02], we learn a probabilistic thesaurus $P_{L\rightarrow L}$, in source language $L$, from the probabilistic bi-lingual dictionaries in L-English $P_{L\rightarrow E}$ and English-L $P_{E\rightarrow L}$. This is equivalent to the 1-step random-walk proposed by us in Chapter 7. Given two words $s_1$ and $s_2$ in source language $L$ and $e$ is a word in English ($E$), $P_{L\rightarrow L}$ is given by:

$$\begin{aligned} P_{L\rightarrow L}(s_2|s_1) &= \sum_{\forall e\in E} P_{L\rightarrow L}(s_2, e|s_1) \\ &= \sum_{\forall e\in E} P_{E\rightarrow L}(s_2|e)\cdot P_{L\rightarrow E}(e|s_1) \\ &\quad \text{(Assuming } s_2\text{, } s_1 \text{ are independent given } e\text{)} \end{aligned}$$

Lexically and semantically related words like morphological variants and synonyms have a high probability score in $P_{L\rightarrow L}$ since they usually map to the same word in the target language. Given a query, we initially run MBF in the source language and let $\Theta_L^F$ be the resultant feedback model. Later, we use the probabilistic thesauri to expand the feedback model as follows:

$$P(f|\Theta_L^{Thesaurus}) = \sum_{\forall s\in S} P_{L\rightarrow L}(f|s)\cdot P(s|\Theta_L^F)$$

The above step includes morphological variants and synonyms for the terms in the feedback model. The final model is obtained by interpolating the $\Theta_L^{Thesaurus}$ with the MBF model $\Theta_L^F$ as shown in Equation 4.2.

For the above experiments, we use the FR-01+02 and DE-01+02 French and German collections. The results of comparison is shown in Figure 5.1. More detailed results are given in Table 5.8. These show that MBF with both thesaurus based expansion and assisting collection in the source language does not perform as well as MultiPRF. MultiPRF automatically combines the advantage of PRF in two different collections and thesaurus based expansion. This addresses the fundamental limitations of MBF and results in an improvement of both retrieval performance and robustness. Another observation we made is that, the quality of the thesaurus learned in this manner is good for French, but poor for German. This can be attributed to the larger vocabulary size in German, along with the lack of a decompounder. Hence the performance of the Thesauri-based method, is much more competitive in the case of French as compared to that in German.

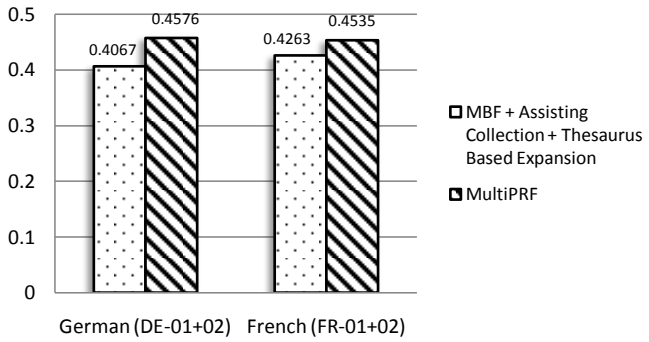


Figure 5.1: MAP score comparison of MultiPRF and MBF with assisting collection in same language and Thesaurus Based Expansion. In MBF experiments, FR-06 and DE-03 were used as assisting collections for French and German respectively.

| Collection | MAP | P@5 | P@10 | GMAP |
|---|---|---|---|---|
| **FR-01+02** | 0.4263 | 0.4682 | 0.4261 | 0.2369 |
| **DE-01+02** | 0.3975 | 0.5205 | 0.4784 | 0.1768 |

Table 5.8: Performance of MultiPRF+Assisting Collection+Thesaurus.

## 5.6 Error Analysis

In this section, we try to analyze the reasons why MultiPRF performs worse than the baseline. On analyzing the results, we found that the following are the major sources of error in our system:

- Poor/Incorrect Query Translation: This is a major source of error in our approach which leads to an inaccurate retrieval in language $L_2$ (e.g. English), thus inevitably drifting the expanded query from its original intent. This also includes terms which could not be successfully translated and do not exist in the English vocabulary *i.e,* OOV terms.
- Poor Feedback Performance in English: In some cases where the right set of terms were not

chosen during translation, the feedback performance is affected thereby reducing the feedback performance in English.

- Query Drift introduced during Back Translation: In a few cases, the back translation too causes a query drift due to the noisy terms it adds. For example on a query whose English translation contains the term "engineering", engineering is stemmed to "engin", which is also the stemmed form of the term "engine". During back translation to $L_1$, we introduce the equivalent translation of "engine" in French "moteur" which is not relevant. Though this problem is taken care of because of other words assisting in disambiguating the intent of the expanded query, this does hurt a few queries.

# Chapter 6

# Extending MultiPRF to multiple assisting languages

## 6.1 Motivation

In the previous two chapters we have seen how our MultiPRF approach, used the assistance of another language to improve PRF in another, by alleviate two of the known problems of PRF, namely Lexical Non-Inclusion and Lack of Robustness. In this chapter, we try to further extend this approach, bu using multiple assisting languages to further improve PRF. Since MultiPRF performed some form of Risk-Minimization by using corpora from two languages, we propose to go further an use more than 1 assisting languages, as the Risk-Minimization principle will still hold, and in fact improve. At the same time, we will still mitigate the problem of Lexical and Semantic Inclusion, as in MultiPRF, because we will still use a probabilistic dictionary to get the feedback terms in the source language. Thus we are motivated towards using more than one language to assist.

## 6.2 MultiAssist PRF

In this approach, we extend the MultiPRF approach, by using more than 1 language to assist. The modified system diagram is given in Figure 6.1. For clarification of notation, refer to Table 6.1, where all notation has been clearly explained. Given a query $Q$ in the source language $L$, we automatically translate the query using a query translation system into each of the assisting languages $L_i$, where $i \in [1, n]$. For assisting Language $L_i$, we repeat the same process as in MultiPRF: Use the translated query to rank the documents and obtain the the top $k$ documents for estimating the feedback model (in language $L_i$) using MBF. Next, the feedback model $\Theta^F_{L_i}$, is translated back into language $L$ using a probabilistic bi-lingual dictionary from $L_i \rightarrow L$, in a similar manner as before.

The final MultiAssistPRF model is obtained by interpolating all the translated feedback models

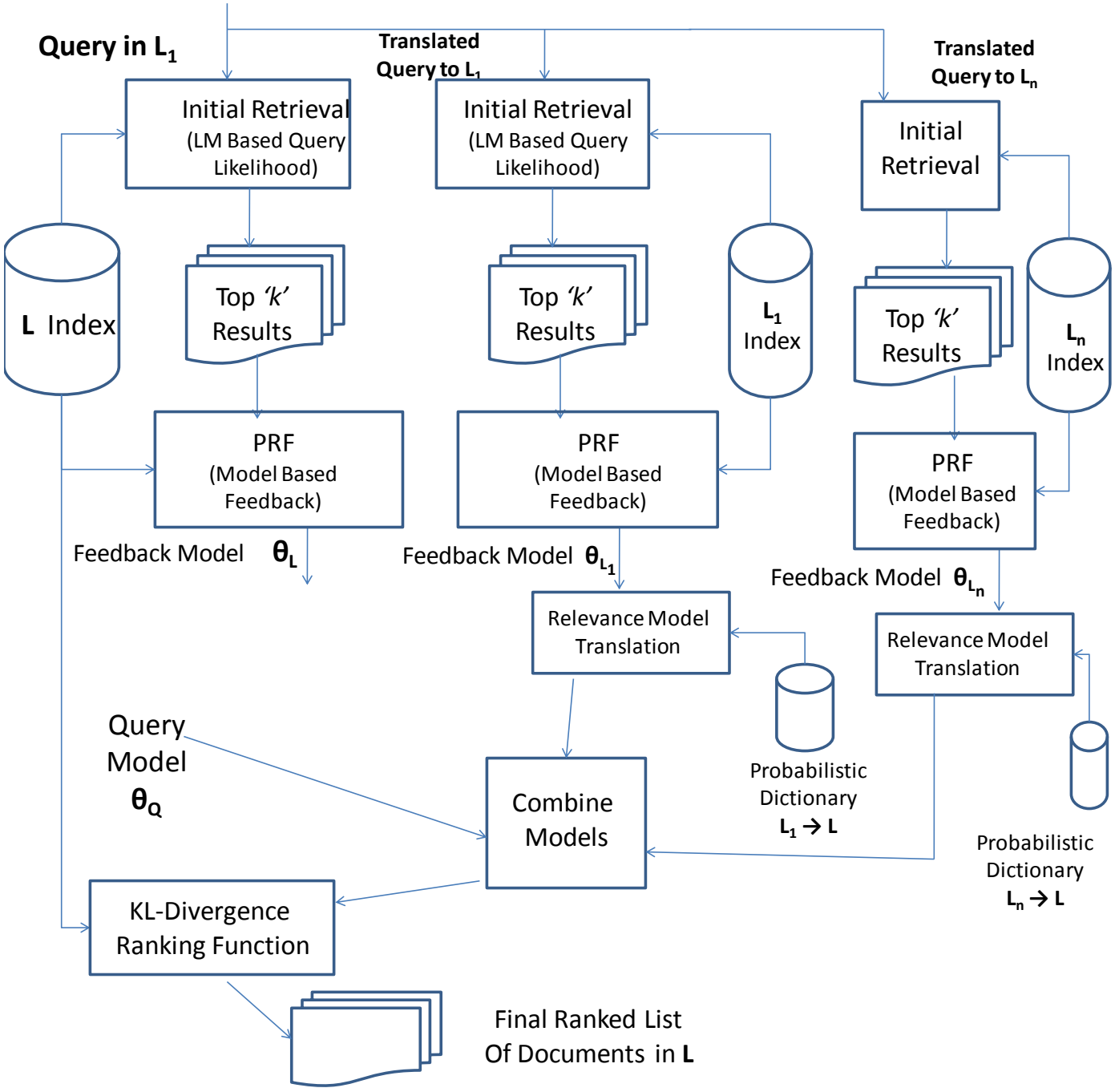


Figure 6.1: Schematic of the MultiAssist Pseudo-Relevance Feedback Approach

| Symbol | Description |
|---|---|
| $\Theta_Q$ | Query Language Model |
| $\Theta_L^F$ | Feedback Language Model obtained from PRF in Source Language $L$ |
| $\Theta_{L_i}^F$ | Feedback Language Model obtained from PRF in Assisting Language $L_i$ |
| $\Theta_{L_i}^{Trans}$ | Feedback Model Translated from $L_i$ to $L$ |
| $\Theta^{MultiAssist}$ | Final Feedback Model computed by MultiPRF |
| $\beta$ | Interpolation coefficients of original feedback model |
| $\gamma_i$ | Interpolation coefficients of feedback model translated from Language $L_i$ |

Table 6.1: Glossary of Mathematical Symbols used in explaining MultiAssistPRF

with the original query model and the feedback model of language $L$ as given below:

$$\Theta^{MultiAssist} = (1 - \beta - \sum_{i \in [1,n]} \gamma_i) \cdot \Theta_Q + \beta \cdot \Theta_L^F + \sum_{i \in [1,n]} \gamma_i \cdot \Theta_{L_i}^{Trans} \tag{6.1}$$

Finally, the documents in the collection of language $L$ are ranked using the final feedback model $\Theta^{MultiAssist}$. The challenge here is the setting of the $\gamma_i$'s, as they then control the relative importance of each of the translation models. In the coming two sections, we discuss two radically different ways of doing this.

## 6.3 Combining Step: Model-1

One way of combining these different model, is to simply give all of them equal weight. This then becomes a direct analogue of the MultiPRF approach, except that now there are multiple assisting languages. We performed experiments with this model, using $n = 2$, *i.e.* Two assisting languages. Hence we have 60 combinations of source and pair of assisting languages. As we see in Table 6.2, in 34 out of the 60 pairs, there was an improvement over the maximum value of MultiPRF (*i.e.* maximum over performing MultiPRF separately, with each of the assisting languages). Thus this itself is a very positive baseline to begin with, as we can further extend this model and hope for more improvements.

## 6.4 Combining Step: Model-2. Selective Weighting

While the previous model (and MultiPRF) for that matter, use constant weights to interpolate, we can do much better, if these weights are adjusted per query. In other words, if for some query, we can figure out that a particular language performs very well, then we will weight that model the highest. Similarly if we can figure out that the original feedback model is doing well, then we need not use assisting languages, since there is a risk of introducing noise via the translation and back-translation steps.

| Source Language | Total Pairs Improved |
|---|---|
| English | 7 |
| French | 6 |
| Spanish | 3 |
| German | 2 |
| Dutch | 8 |
| Finnish | 8 |
| Total | 34 |

Table 6.2: Number of Pairs where MultiAssistPRF improves over the Max Value of MultiPRF

However making this selection is a non-trivial step. One method of doing this is by using a classifier, to learn relative weights of the different models. Alternatively, we could learn a decision tree, which as its' alternatives could suggest using only 1 particular model, or 2 models or all 3 models. However for either of these options, we would have to come up with some features, based on which this selection happens. We list some potentially useful features:

- *Statistics of Translated Query Words in Top k*: Some potentially useful features would be statistics such as mean and variance of total counts of the query words (of the translated query), in the documents of the feedback set in the assisting language collection. More the query words, the more confident you can be of the documents being relevant.
- *KL-Divergence of Feedback Model from the translated Query Words*: Another feature which could help in this task is the feedback model obtained by the MBF algorithm, and its' divergence from the Query Model of the translated Query.
- *KL-Divergence of Translated Feedback Model from the Original Query Words*: Another potentially potent feature is the divergence between the translated feedback model (*i.e.* after performing the back-translation), and its' divergence from the Original Query Model.
- *KL-Divergence of Top 100 Documents MLE Model from the translated Query Words*: Another set of features which could help is the divergence of the MLE model of the Top 100 Documents (both before and after feedback) from the Query Model of the translated Query.
- *KL-Divergence of Top 100 MultiPRF Documents MLE Model from the Original Query Words*: Another feature which is likely to help is the divergence of the MLE model of the Top 100 Documents of the MultiPRF model using that assisting language from the Query Model of the Original Query.

Thus once we train a classifier/decision tree using these features, we will be able to assign weights to the different models, based on the query. Preliminary results, which supposed the existence of such as ideal classifier, for the case of $n = 2$, were able to get MAP improvements of the order of 15% over the MBF baselines, and 7-8% over the maximum of the two MultiPRF values.

# Chapter 7

# Random Walk Across Languages

As seen in previous chapters and representative queries, the addition of synonyms and morphological variants to the query, along with co-occurrence based terms, can lead to improvements. The methods seen previous to this, all perform Pseudo-Relevance Feedback, by modifying the baseline algorithm. In this chapter we propose methods of accumulating lexically and semantically related variants of query terms, without necessarily performing PRF. Though such a method is not expected to perform as well as previous methods, we still expect it to be competitive with respect to other synonym-based expansion methods. As mentioned previously, it has been reported in literature that directly using resources such as WordNet, for obtaining these terms for expansion, leads to negative results. Thus the method, we propose uses an alternate resource: a *probabilistic thesaurus.*

In previous chapters of this thesis, we have seen the expressive power of the Language-Modeling Framework. Due to strong extensions to the basic model, along with powerful smoothing methods proposed, this framework is extremely attractive to use. Hence we use the LM framework to formulate our method. The idea is simple: Given a query $Q$, with query terms $(q_1, q_2, q_3, ..., q_n)$, we find expansion terms $(t_1, t_2, .....t_m)$, which maximize the probability $P(t_1, t_2, ...., t_m|q_1, q_2, ...., q_n)$. We will discuss ways of finding this probability by performing random walks across languages and some relaxations to the model we make for simplicity.

## 7.1 Model-1

Here we discuss a very simple model, which is similar to the one proposed by [XFW02]. In this method we use a statistical thesaurus to perform expansion. Suppose we are given a statistical thesaurus, with values $P(a|b)$, indicating the probability that $a$ is a synonym of $b$. We thus find an expansion model $\theta_E$ of the original query $Q$ as:

$$P(w|\theta_E) = \sum_{q_i \in Q} P(w|q_i)$$

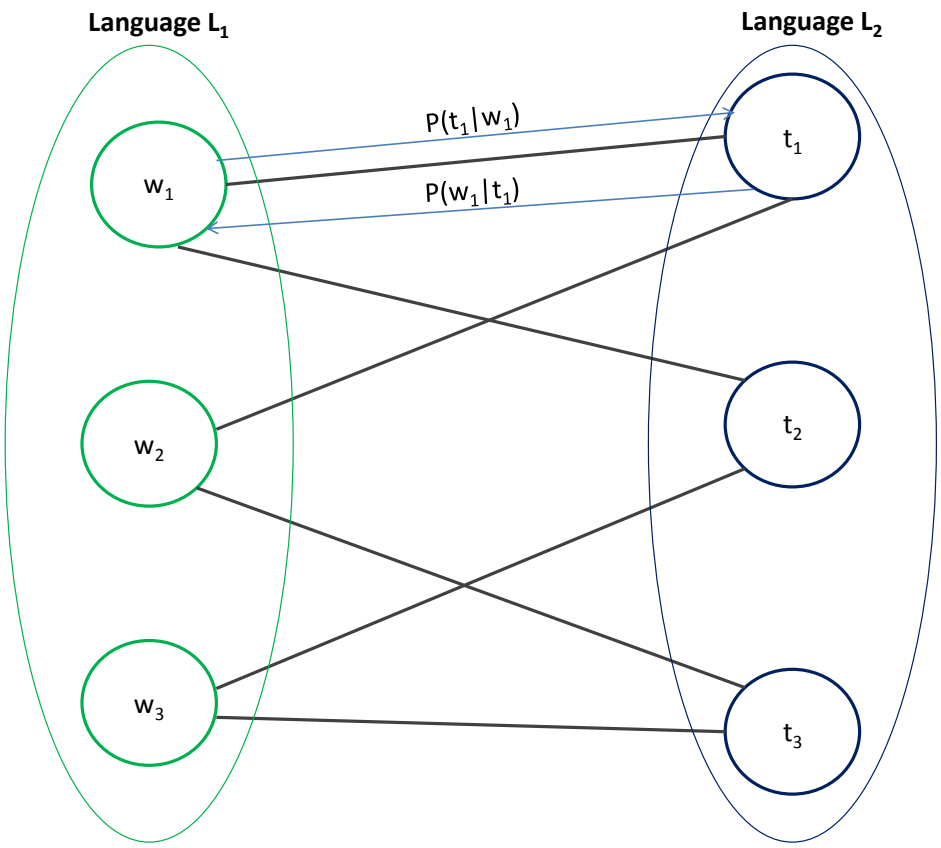


Figure 7.1: Random Walk Across Languages

This seems like a very intuitive method of performing expansion as well. In earlier chapters, we contrasted MultiPRF with this as a synonym-based expansion method. The problem now changes to how we can obtain such a thesaurus. Probabilistic thesauri are a very potent resource, and find applications in many fields of NLP. However in practice, these are very rare as they are extremely difficult to create. We this propose a method of creating these thesauri, by using probabilistic bilingual dictionaries (which are easily available and simple to create) between two the source language $L_1$ and a *friend language* $L_2$. In our method, we perform a random walk across terms from one language to terms in another language using the edge weights as given in the probabilistic bilingual dictionaries. This can be seen in Figure 7.1. To go from a term $s$ in the source language to a term $t$ in the friend language, the weight of the edge will be $P(t|s)$, which can be found in the bilingual dictionary. For the reverse path, the probability will be $P(s|t)$.

Thus is we were to perform 1-step of this random-walk, we obtain a thesaurus with probabilities $P_1(w_2|w_1)$ as:

$$P_1(w_2|w_1) = \sum_{t_i \in Vocab(L_2)} P(w_2|t_i).P(t_i|w_1)$$

Thus we obtain the same formulation as that of Xu et. al. However our framework, allows us to go further as well, where we use more than 1-step. For example, the 2-step random walk

| ORIGINAL FRENCH STEM | MEANING | THESAURUS-BASED SYNONYM | MEANING | PROBABILITY |
|---|---|---|---|---|
| parfum | Perfume/ Fragrance | odorifer | Sweet Odor | 0.1833 |
| | | fragranc | Fragrance | 0.1814 |
| | | parfum | Perfume | 0.1480 |
| | | flavour | Fragrance | 0.1279 |
| | | aromat | Aroma | 0.029 |
| | | arom | Aroma | 0.024 |
| foudr | Lightning/ Expression of Anger | corroux | Anger | 0.1 |
| | | foudr | Lightning | 0.0756 |
| | | coler | Anger | 0.0108 |
| | | encour | Anger | 0.006 |
| | | anger | Anger | 0.006 |

Table 7.1: Top "Synonyms" for a few words as in the 1-step French Thesaurus

probabilities can be obtained as:

$$P_2(w_2|w_1) = \sum_{w' \in Vocab(L_1)} P_1(w_2|w').P_1(w'|w_1)$$

Thus the n-step walk probabilities will have the general form of:

$$P_n(w_2|w_1) = \sum_{w' \in Vocab(L_1)} P_{n-1}(w_2|w').P_1(w'|w_1)$$

### 7.1.1 Experiments and Results

We ran this algorithm on the French and German collections. In both of these cases we used *English* as the *friend language*, due to observed similarities between the languages. As mentioned before the probabilistic bi-lingual dictionary $t(f|e)$ is learned from a parallel sentence-aligned corpora in $L_1 - L_2$ based on word level alignments. As before these were learnt automatically by running GIZA++ [ON03] on a parallel sentence aligned corpora provided by *Europarl* [Phi05].

For the two languages, in Tables 7.1 and 7.2, we find examples of the entries in the learned 1-step statistical thesaurus along with the probabilities associated with these translations. These show the learned thesaurus to be sensible, and able to find synonyms. On deeper analysis, we found the French thesaurus to be more accurate than the German thesaurus. In other words, the French

| ORIGINAL GERMAN STEM | MEANING | THESAURUS-BASED SYNONYM | MEANING | PROBABILITY |
|---|---|---|---|---|
| akteur | Action/Actor | akteur | Actor | 0.1606 |
| | | beteiligt | Involved | 0.031 |
| | | spiel | Played | 0.0242 |
| | | haupakteur | Main Actor | 0.013 |
| hund | Dog | hund | Dog | 0.0178 |
| | | beiss | Bite | 0.006 |
| | | hetzjagd | Chase | 0.002 |
| beitritt | Accession | mitgliedschaft | Member | 0.016 |
| | | beitritt | Accession | 0.008 |
| | | mitgliedstaat | Member | 0.003 |
| | | eu-mitgliedschaft | EU-Member | 0.003 |

Table 7.2: Top "Synonyms" for a few words as in the 1-step German Thesaurus

thesaurus assigns high probabilities to words which are synonyms, while the German thesaurus placed much smaller probability mass on such terms, and also contained lot of noisy terms. This can be seen from the examples presented as well. Another observation we made, was with respect to frequent terms which have multiple senses. In such cases the quality of the thesaurus entries seemed to be poor as well. We also observed more related terms coming up, as we increases the number of steps from 1-step to 2-step, but since synonyms are the most important terms for us, the 1-step thesaurus suffices.

Once we obtain these thesauri we can use them in the query expansion algorithm proposed above. Though we do not expect to perform as strongly as its' PRF counterparts, we still expect it to perform comparable to Synonym Expansion Methods. Since this was the simplest model, we can expect to improve with extensions of this method. We discuss some of these in the next section.

## 7.2 Possible Extensions

As noted, Model-1 was too simple. One of the flaws observed in Model-1, was that it found synonyms for query terms independent of the other query terms. This however is suboptimal, as for some ambiguous terms, the context around them (*i.e.* the words co-occurring with it) help disambiguate the correct sense. For example, while the word "*bank*" is ambiguous in itself, when preceded by the word "*river*" the word is disambiguated. Thus we expect the synonyms of "*bank*" to be different when it occurs in "*river bank*" as compared to when it appears in "*financial bank*". Hence while Model-1 naively assumes independence of synonyms of each of the query words, we

can fix this problem easily in different ways:

- When going from query term $q'$ in Language $L_1$ to a term $t'$ in Language $L_2$, we only consider those $t'$, which are at some distance $k$ from another query term. This forces only those terms to be used, which are at least remotely related to some other query term. This is a very simple way of enforcing dependence between query terms
- Alternately while selecting expansion terms we only select those terms which have the $K$ highest probability scores ($\sum_{q_i \in Q} P(w|q_i)$) associated with them. To make this a valid probability, we normalize after selecting the top $K$. Thus here the equation becomes $P(w|\theta_E) = P(w|q_1, q_2, ..., q_n)$ as terms which get combined evidence from multiple query words will be chosen as expansion terms. This can again be combined with the above modification as well, to get a more realistic model.

# Chapter 8

# Pseudo-Irrelevant Documents

## 8.1 Motivation

In Information Retrieval the goal is to obtain documents relevant to the query issued. Most methods return a ranked list of documents, containing both relevant and irrelevant documents. While an ideal ranking would return all the relevant documents before the irrelevant documents, in practice the two sets are interspersed in the ranked list, with some irrelevant documents ranked above relevant documents and vice-versa. Hence the onus now shifts on the user to identify which of these are relevant or not. In web-search the snippet of a webpage assists the user in this task of identifying documents matching their intent.

During a web search for a query, the user tends to click on a few documents which he believes could be relevant. In many scenarios, some of the documents the user clicked on, may not be relevant. However there are some documents which the user reads the snippet of and chooses to ignore. These documents inevitably are irrelevant and do not match the user's intent. Hence human beings may not be able to identify relevant documents perfectly, but they can identify some (highly-ranked) irrelevant documents much more easily. This motivated us to ask the question, if machines could do the same. While most IR engines try to identify documents by their relevance, we explore the possibility of identifying documents by their irrelevance, which is an easier task for a human.

### 8.1.1 Ranking as per Irrelevance

We look at this in a more formal manner. Let $q$ be the query, $C$ be the collection and $D \in C$ be a document in the collection. The ideal relevance function $r$ assigns a score $r(D)$ to each document $D \in C$, depending on how well $D$ models the information need of $q$. Analogous to this we have an ideal irrelevance function $ir$, which assigns score $ir(D)$ to each document $D \in C$. We can write this as:

$$ir(D) = 1 - r(D)$$

Hence if we get $r(D)$ we get $ir(D)$ as well, and vice-versa. In the Language-Modelling framework we approximate $r$ as $r_{lm}(D) \equiv P(q|D)$. Since this is an approximation, $ir_{lm}(D) = 1 - r_{lm}(D) =\neq ir(D)$. Now consider a new framework where instead we approximate $ir(D)$ as $ir'(D)$ which is not dependant on $r_{lm}(D)$. Corresponding to this we similarly have $r'(D)$.

Since language-modelling is known to do well in identifying relevant documents, it is fair to assume that $r_{lm}$ is a better approximation to $r$ than $r'$. By better approximation we mean the precision of $r_{lm}$ is better than $r'$. Now we try to find an $ir'$ which is a better approximation of $ir$ than $ir_{lm}$. Note that both of these can hold, i.e. good precision in relevance need not lead to good precision in irrelevance. Consider the example of a collection of 100 relevant documents and 100 irrelevant documents. Suppose $r_{lm}$ has 100% precision but recall of only 10, then precision of $ir_{lm}$ is only 100/190. Hence it is possible to find such an $ir'$. The idea is that if we can obtain a high-precision irrelevant function $ir'$, we can eliminate irrelevant documents which are ranked high by $r_{lm}$ thus improving precision of $r_{lm}$.

The idea mentioned above can be exploited directly, as we will see in Chapter 10. However the key point from the above formulation is the fact that we are able to improve PRF, by identifying high-scoring irrelevant documents. Thus this motivates us to find these documents.

## 8.2 Pseudo-Irrelevant Documents

### 8.2.1 High-Scoring Irrelevant Documents

High-Scoring Irrelevant Documents can be classified in a couple of ways. The first classification is based upon the ease of identifying them: they are either a)Easy to Identify; b)Difficult to Identify. For example in the websearch example, the irrelevant pages the user ignores are easy to identify; while the pages he clicks and later finds irrelevant are more difficult to find. The method proposed in section 8.1.1 and later elucidated in chapter 10 is a high-precision method, where a few irrelevant documents are identified. These documents which can be identified in this manner(using a high-precision irrelevance classifier) are referred to as Easy to identify. However we will still be left with other irrelevant documents which are Difficult to identify.

The second kind of classification is based on where they occur: the irrelevant document is either in the feedback set, or it is outside the feedback set. Irrelevant documents in the feedback set are mistakenly considered as pseudo-relevant, due to the assumption PRF, and hence have the most adverse effect. Identifying them is detailed in 10. In this chapter we focus on identifying high-scoring irrelevant documents outside the PRF set, i.e. irrelevant documents which are ranked in the top $N$(high-scoring) but are outside the top $k$(feedback set).

Though retrieval performance is not adversely affected by these documents as by the irrelevant documents in the feedback set, the presence of these documents, causes relevant documents to be ranked below them. In other words, an irrelevant document at rank $r$ will affect performance if

there are any relevant documents below rank $r$ as then they have been ranked below an irrelevant document. In Table 8.1, we see the large performance increases which are possible if these irrelevant documents are identified with $N = 100$ and $k = 10$. Refer to Appendix A for details of the experiments performed, and collections used.

In columns 2-4 we have the performance of the simple LM algorithm, in columns 5-7 we see the performance improvement on simply identifying and removing the irrelevant documents from the LM ranking. In the last column we have the average number of irrelevant documents removed per query. We see how performance increase varies from 25-50% across collections, on doing this. Though identifying these irrelevant documents may not be easy, because of the large performance increase seen, the hope is that identifying even some of the irrelevant documents can lead to large performance increase.

| Coll. ID | LM | | After Elimination | | % MAP Change | Avg. Irrel. Docs. Removed |
|---|---|---|---|---|---|---|
| | MAP | P@100 | MAP | P@100 | | |
| LAT | 0.4338 | 0.1181 | 0.5361 | 0.1332 | **+23.58%** | 81 |
| GH | 0.3819 | 0.1573 | 0.493 | 0.1849 | **+29.09%** | 78 |
| AP | 0.2772 | 0.2655 | 0.3809 | 0.3287 | **+37.41%** | 68 |
| WSJ | 0.266 | 0.2611 | 0.367 | 0.3193 | **+37.96%** | 68 |
| SJM | 0.2074 | 0.1572 | 0.3019 | 0.1931 | **+45.56%** | 76 |

Table 8.1: Performance Increase on Eliminating Irrelevant Documents between ranks $11 - 100$

Also, identifying these irrelevant documents could help us in identifying the irrelevant documents in the feedback set, since the irrelevant documents identified will be more similar to the irrelevant documents in the top $k$ rather than the relevant documents in the top $k$. Alternately if we can identify some of these irrelevant document, we could use them as negative feedback in techniques such as the Rocchio Model, or the Distribution Separation Method by Zhang et.al.

### 8.2.2 Defining Pseudo-Irrelevance

Identifying the high-scoring irrelevant documents is not too easy, so we need some approximation. We are inspired by the idea of pseudo-relevance which tries to approximate a relevant set of documents by using the top $k$ documents. Hence analogous to this we define the concept of pseudo-irrelevant documents.

*Pseudo-Irrelevant Documents* are high-scoring (inside top $N$) documents, that are highly unlikely to be relevant. Assuming the top $k$ documents to be relevant(PRF assumption), we identify documents in the top $N$ which are highly dissimilar to the top $k$ documents, and thus unlikely to be relevant. Since these documents are dissimilar to the pseudo-relevant documents we call such documents pseudo-irrelevant. In the next section we see a more formal description of their identification.

## 8.3 Identifying Pseudo-Irrelevant Documents

We propose a method to find pseudo-irrelevant documents wherein we find high-scoring documents, which are highly dissimilar to the pseudo-relevant documents. Let $D_N$ be the set of top $N$ documents retrieved in the first-pass retrieval, i.e. the high-scoring documents. Let $D_k = d_1, d_2, ...d_k$ refer to the pseudo-relevant documents i.e. the top $k$ documents from the first-pass retrieval. Let $T_{d_i,m}$ be the set of $m$ documents that are most similar to document $d_i; i \in ([1, ..., k]$. Thus the set $T_m = \cup_{i=1}^{i=k} T_{d_i,m}$ refers to the set of documents that are are similar to the feedback documents, i.e. the pseudo-relevant documents. Pseudo-Irrelevant documents are those documents that are:-

1. High-Scoring.
2. Dissimilar to the feedback documents.

Thus using this we can obtain the Pseudo-Irrelevant Set $P_I$ as :

$$P_I = D_N - (D_N \cap T_m)$$

**Obtaining $T_{d_i,m}$:** To obtain $T_{d_i,m}$ consider the following method: We construct a query out of the document $d_i$ and find the highest ranking documents. Thus we create a query $Q_{d_i}$ from $d_i$ suing the terms in it, excluding the outliers[1]. Using $Q_{d_i}$ we get the top $m$ documents as an estimate of $T_{d_i,m}$.

**Algorithm 1** : GetPseudoIrrelvantDocuments(N,k,m)

```
Rank the documents using first-pass retrieval scheme.
Obtain D_N and D_k = d_1, d_2, ...d_k.
Initialize T_m = φ.
for Every d_i ∈ D_k do
    Convert d_i to Query Q_{d_i}.
end for
for Every Q_{d_i} do
    Parallelly rank documents in collection using the Q_{d_i}s.
end for
for Every Q_{d_i} do
    Obtain top m of each ranking as T_{d_i,m}.
    T_m = T_m ∪ T_{d_i,m}.
end for
Return D_N − (D_N ∩ T_m).
```

Thus putting everything together we get the algorithm for identifying pseudo-irrelevant documents. Though the algorithm is based on the assumption that the pseudo-relevant documents are indeed relevant, it is robust if it fails to hold as well. The main principle behind this are :

[1]Outliers are terms which either appear in $\leq 5$ documents in the collection(too few occurrences); or have IDF $\geq \log 10$ (too many occurrences)

*Relevant documents are more similar to each other, than Irrelevant Documents.* This is because while the relevant documents are about the same topic, irrelevant documents differ in topics. Hence a relevant document in the top $k$ will have other relevant documents which are similar to it in $D_N$, while irrelevant documents in the top $k$ may not have many such documents. Hence the residue after running the algorithm is a cleaner set, i.e. a set with a much larger % of irrelevant documents.

## 8.4 Accuracy of Identification Method

We know evaluate the performance of our method. Ideally we would want the system to remove all the relevant documents from $D_N$, to obtain $P_{opt}$. However this is a difficult task and thus we evaluate for % accuracy of the pseudo-irrelevance set, i.e. the fraction of the pseudo-irrelevant set which is irrelevant, and irrelevance recall i.e what fraction of the optimal set of documents from $P_{opt}$ is there in $P_I$. We use $N = 100$ in our experiments, and modify $m$ accordingly. Intuitively we feel that increasing $m$ should increase accuracy and decrease recall, which we confirm in the results shown in Table 8.2. Column 2 in the table refers to the percentage of irrelevant documents the original $D_N - D_k$ set after the first pass retrieval. Size refers to the average size of the pseudo-irrelevant set returned.

| ID | Base Acc% | For $m = 10$ | | | For $m = 20$ | | | For $m = 30$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc% | Recall | Size | Acc% | Recall | Size | Acc% | Recall | Size |
| LAT | 90.8% | 93.0% | 92.2% | 80 | 94.0% | 86.4% | 74 | **94.6%** | 82.8% | 70 |
| GH | 86.6% | 88.8% | 91.8% | 80 | 89.9% | 86.4% | 74 | **90.7%** | 82.7% | 71 |
| AP | 75.4% | 77.8% | 91.5% | 79 | 79.3% | 85.9% | 72 | **80.1%** | 81.8% | 68 |
| WSJ | 76.0% | 78.1% | 92.5% | 80 | 79.3% | 86.6% | 74 | **79.9%** | 82% | 69 |
| SJM | 85.7% | 87.7% | 90% | 78 | 88.7% | 84.2% | 72 | **89.5%** | 80% | 68 |

Table 8.2: Change in Algorithm performance with $m$

Thus we see as expected the accuracy of identification increases as $m$ increases.The accuracy values differ across collections, depending on the baseline of the collection with a **4%-8%** improvement over the baseline varying across collections. These can be further improved but then recall is compromised. Another observation we make is that the recall values and average size of the pseudo-irrelevant set are almost identical, which indicates that the method is quite robust and gets good recall values even in more difficult collections.

Since the algorithm works on the assumption that the pseudo-relevant set is actually relevant, we see how the performance varies with the number of relevant documents in the top $k$. In table 8.3, we see how the performance depends on the number of relevant documents in the top $k$. Another trend we saw was that: Queries with many relevant documents in the PRF set, inevitably had a lot of relevant documents in $D_N$ also. Hence the baseline itself for these queries is quite low and though our method improves on this value it is not as high as on other queries. Hence we felt a better estimator of performance in this case is the improvement over baseline rather than the absolute value. We have done a query-wise analysis of the AP and LAT collections, and provided the average

% change in accuracy for $m = 10/20/30$. As we see the percentage improvement improvement in accuracy over the baseline increases for all the queries.

| NumRel ∈ PRF | %Change in Accuracy | | | | | |
|---|---|---|---|---|---|---|
| | AP | | | LAT | | |
| | $m = 10$ | $m = 20$ | $m = 30$ | $m = 10$ | $m = 20$ | $m = 30$ |
| 0-1 | 0.7% | 1.2% | 1.4% | 0.2% | 0.3% | 0.3% |
| 2-3 | 1.2% | 1.4% | 2% | 1.7% | 2.4% | 2.5% |
| 4-6 | 3.2% | 6.4% | 8.9% | 3% | 4.3% | 5.3% |
| 7-8 | 6.5% | 10.5% | 12.7% | 4.7% | 6.9% | 8.3% |
| 9-10 | 14.3% | 25.3% | 30.6% | 4.8% | 7.4% | 9.7% |

Table 8.3: Dependence on number of relevant documents in the top $k$

We see consistent % improvements as the number of relevant documents in the PRF set increases. At the same time even when there are a few relevant documents in the PRF set, our method leads to an improvement over the baseline in both collections, thus highlighting the robustness of our method. Thus in this chapter we have presented our method of extracting a set of pseudo-irrelevant documents and displayed its' performance over different collections, as well as analyzing its' performance on a query-wise level as well.

# Chapter 9

# Using Pseudo-Irrelevance Data

In the previous chapter we have seen how to obtain a reasonably large set, which majorly consists of irrelevant documents. This information can now be utilized in many ways. For example:

1. The set of pseudo-irrelevant documents can be used as irrelevant data, in Rocchio's method
2. This set or a subset of it can be used as the seed irrelevance distribution in Zhang et.al 's Distribution Separation Method.
3. Can be combined with any other Negative Feedback technique, as the set consists of primarily high-scoring irrelevant documents.

In this chapter we propose a couple of methods to use this pseudo-irrelevance data to improve PRF. In the first approach, we try to obtain a feedback model consisting of a small set of "good" discriminative terms, which are terms which help us differentiate between the pseudo-relevant documents ($+ve$ examples) from the pseudo-irrelevant documents ($-ve$ examples). In the second method, we treat the pseudo-irrelevance set, as a mixture model of the collection model and a query-specific noise model. We then treat the PRF set also as a mixture of this query-specific noise and of a relevance feedback model. We then use the EM algorithm twice to finally obtain the relevance feedback model.

## 9.1 Obtaining Discriminative Terms

### 9.1.1 Key Idea

The main idea behind query expansion is that though the queries are short, there exist additional terms in the relevant documents, which are *discriminative* and can help differentiate between relevant and irrelevant documents, thus leading to a performance improvement when these terms are added to the query. If we were given the set of relevant documents and the set of irrelevant

documents, then if we learnt a binary classifier on the combined set with the relevant and irrelevant documents provided as $+ve$ and $-ve$ examples respectively, with terms as features, then the terms with highest positive feature weights, are the most discriminative terms, as they help separate the relevant from the irrelevant documents.

### 9.1.2 Our Method:Discriminative PIR or PIR1

The above idea is the key idea behind our method. Since we do not have either relevant or irrelevant documents available to use, we make use of the pseudo-relevant documents as the relevant ($+ve$ examples) and the pseudo-irrelevant documents as the irrelevant ($-ve$ examples). We then train a logistic classifier on this set, with one feature for every non-outlier term in the feedback set. The feature value for a term in a document is its' $tf - idf$ value in the document. Once the classifier is trained we then use the learned weights and pick the terms with the highest feature weights as the expansion terms.

These terms are good expansion terms, because they increase the scores of the top $n$ documents(which are assumed to be relevant) while at the same time they do not increase the scores of the pseudo-irrelevant documents(which are assumed to be high-scoring irrelevant documents). If a term has a $+ve$ feature weight then its' presence in a document contributes to the document being classified as $+ve$, while terms with $-ve$ feature weights contribute to a document being classified as $-ve$.

To put the methid formally, we train a logistic classifier with $D_k$ as $+ve$ examples, and $P_I$ as $-ve$ examples. As features we use the non-outlier terms in these documents, with feature values being the $tf - idf$ score. The linear discriminant function learnt $w$ is then sorted as per feature weights, and the top 20 feature values are picked and a language model of the corresponding terms created with uniform weights to all terms. This feedback model is then interpolated with the initial query in the ratio of $\alpha : (1 - \alpha)$.

**Algorithm 2** : UsePseudoIrrelvantDocumentsPIR1($P_I$,$D_k$,$\alpha$)

1: Create a Feature Table of the documents in $P_I \cup D_k$.
2: Fill up using a term's feature value as $tf - idf$ value.
3: Set the Label vector as documents in $P_I$ as $-1$ and those in $D_k$ as $+1$.
4: Run Logistic classifier and Learn weight vector $w$.
5: Sort the weight vector in descending order
6: Choose top 20 Terms as $T'$
7: Create Language Model $\theta_F$ as uniform model over $T'$
8: Get interpolated model $\theta' = \alpha\theta_F + (1 - \alpha)\theta_Q$

Above algorithm completely describes the main steps in the **PIR1** method, of using pseudo-irrelevance information, while the next subsection describes an analyzes the performance.

### 9.1.3 Experimental Results

We tested our method on different collections. We used LibLinear(Appendix A.1.3) as the logistic classifier with default settings used. We use $k = 10$ and obtain pseudo-irrelevant documents as described in previous chapter with $N = 100$ and $m = 10/20/30$. We interpolate the final model learnt using interpolation co-efficient $\alpha = 0.4$. We give the results for different collections in Table 9.1.

| ID | LM | | MF | | PIR1: $m = 10$ | | PIR1: $m = 20$ | | PIR1: $m = 30$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MAP | P@5 | MAP | P@5 | MAP | P@5 | MAP | P@5 | MAP | P@5 |
| LAT | 0.434 | 0.485 | 0.442 | 0.5 | **0.472** | 0.503 | 0.471 | 0.503 | 0.472 | 0.499 |
| GH | 0.382 | 0.424 | 0.405 | 0.434 | **0.429** | **0.474** | 0.43 | 0.471 | 0.427 | 0.47 |
| AP | 0.277 | 0.468 | 0.327 | 0.495 | 0.328 | **0.517** | 0.33 | 0.513 | 0.33 | 0.517 |
| WSJ | 0.266 | 0.483 | 0.303 | 0.516 | 0.305 | **0.527** | 0.305 | 0.529 | 0.306 | 0.529 |
| SJM | 0.207 | 0.34 | 0.235 | 0.364 | 0.236 | 0.36 | 0.239 | 0.357 | 0.239 | 0.353 |

Table 9.1: Change in Feedback performance with $m$

As can be seen from the values, the method performs almost uniformly for differing values of $m$. This can be explained using the recall-precision tradeoff which occurs on changing $m$. Another aspect to note is that though the $MAP$ values increase in all collections the magnitude is smaller $1-2\%$ while in the CLEF collections it is much larger: $6-7\%$. At the same time, the $P$@5 values increase considerably in the TREC collections. The explanation for the results on the TREC collections is that though the method is getting high $P$@5 values, the ranking of the documents lower down the list is sub-optimal. This can be attributed to the lower accuracy of the pseudo-irrelevant set in these collections, due to which the weights learnt are not optimal.

Thus we see how the PIR1 method, performs consistently across collections beating the baseline values in all collections. We also feel that the method has further potential for improvement and we continue to work in this direction.

## 9.2 Query-Specific Noise Separation

### 9.2.1 Key Idea

In Model-Based Feedback, the feedback document set is assumed to be a mixture model of collection noise and a feedback model. However we feel that the noise is not the same across all documents. We feel there is some irrelevant data noise which is introduced into the feedback set(due to the presence of irrelevant documents), apart from the collection noise. This is irrelevant data noise can be obtained from the topics of other high-scoring irrelevant documents. Hence we propose to first use the (pseudo) irrelevant documents to obtain the irrelevant-noise model(or the query-specific

noise). Next we use a similar idea to the Model-Based Feedback, but instead use the irrelevance-noise model instead of the collection noise model, thus obtaining a cleaner version of the relevance feedback model.

### 9.2.2 Our Method: Noise-Separating PIR or PIR2

This method is intended to be an extension to the Model-Based Feedback method. While there the feedback set is assumed to be a linear combination of collection noise, and a feedback model, we believe the noise is not the same for all queries and there is a query dependent factor, which mainly originates from the irrelevant documents in the top $k$. Thus we propose to find this noise by extracting a query-specific noise model $\theta_N$ from the pseudo-irrelevant documents. We assume that the pseudo-irrelevant documents are a mixture if the collection noise and the query-specific noise mode $\theta_F$, or in other words the pseudo-irrelevance distribution $\theta_{P_I}$ is a linear combination of the collection noise and $\theta_N$:

$$\theta_{P_I} = \lambda_1 \theta_C + (1 - \lambda_1)\theta_N$$

We obtain $\theta_N$ using EM updates similar to the one in the model-based feedback, and thus obtain a model of $\theta_N$. Now the feedback set contains irrelevant documents which tend to be topically related to the high-scoring irrelevant documents of the pseudo-irrelevant set. Hence we propose that the feedback document distribution $\theta_{D_k}$ is a linear combination of the query-specific noise $\theta_N$ and a relevance feedback model $\theta_F$.

$$\theta_{D_k} = \lambda_2 \theta_N + (1 - \lambda_2)\theta_F$$

We again use the EM algorithm, with the same update equations to obtain $\theta_F$. We then prune terms with very small weights as in Model-Based Feedback. Once we obtain $\theta_F$ we interpolate with initial query model with co-eficient $\alpha$ to obtain the final model. Note that in this method too we ignore outlier terms, and hence all models are created excluding such terms.

**Algorithm 3** : UsePseudoIrrelvantDocumentsPIR2($P_I$,$D_k$,$\alpha$)

1: Obtain $\theta_C$, and MLE estimate of $\theta_{P_I}$
2: Use the EM algorithm to solve and obtain $\theta_N$
3: Obtain MLE estimate of $\theta_{D_k}$
4: Use the EM algorithm to obtain $\theta_F$
5: Prune terms from $\theta_F$ whose value $\leq 0.001$ and then re-normalize $\theta_F$.
6: Get interpolated model $\theta' = \alpha\theta_F + (1 - \alpha)\theta_Q$

Above algorithm completely describes the main steps in the **PIR2** method, of using pseudo-irrelevance information, while the next subsection describes an analyzes the performance of the algorithm.

### 9.2.3 Experimental Results

We tested the PIR2 method as well on different collections. Here too to obtain pseudo-irrelevant documents, we use the algorithm described in previous chapter with $N = 100$ and $m = 30$. We used the value of $\lambda_1$ as 0.5 and $\lambda_2$ as 0.4. We interpolate the final model learnt using interpolation co-efficient $\alpha = 0.3$. We use 30 EM iterations for both phases as we have observed that is generally sufficient to lead to good convergence.We give the results for different collections in Table 9.2.

| ID | LM | | MF | | PIR2 | | |
|---|---|---|---|---|---|---|---|
| | MAP | P@5 | MAP | P@5 | MAP | P@5 | %Change from MF |
| LAT | 0.434 | 0.485 | 0.442 | 0.5 | 0.452 | 0.493 | **2.2%** |
| GH | 0.382 | 0.424 | 0.405 | 0.434 | 0.419 | 0.455 | **3.5%** |
| AP | 0.277 | 0.468 | 0.327 | 0.495 | 0.339 | 0.503 | **3.7%** |
| WSJ | 0.266 | 0.483 | 0.303 | 0.516 | 0.311 | 0.533 | **2.7%** |
| SJM | 0.207 | 0.34 | 0.235 | 0.364 | 0.243 | 0.379 | **3.4%** |

Table 9.2: Performance of algorithm PIR2

As can be seen from the values, the method is more robust than PIR1 with 2-4% performance increase across all collections. The % change given in the last column indicates the significant improvements. Also the $P@5$ value is also seen to rise considerably across collections. All this indicates that the method is highly robust and indicates that the proposition of a query-specific noise aspect in addition to the collection noise.

### 9.2.4 Future Improvements

The values given in Table 9.2 are just preliminary results. We yet have to do an exhaustive search over the parameter space to find the optimal parameter values for $\alpha, \lambda_1$ and $\lambda_2$. Also the current model does not add a term for the collection noise while calculating $\theta_F$. We believe there is definitely a collection-noise aspect also in the feedback set, as Zhai,Lafferty proposed. Thus the equation would become

$$\theta_{D_k} = \lambda_2\theta_C + \lambda_2\theta_N + (1 - \lambda_2 - \lambda_3)\theta_F$$

We believe that using this equation with a small value of $\lambda_3$(approx. 0.05-0.15) would further improve the model, as the current model does not account for the collection noise in the feedback set. Hence we feel due to all these factors that the performance of the PIR2 algorithm can be further improved from the current levels, it is at.

# Chapter 10

# Identifying Irrelevant Documents in the PRF set

## 10.1 Motivation

Pseudo-Relevance Feedback is based on the assumption that the top $k$ documents(which constitute the feedback set) are relevant. However this assumption rarely holds in practice, as there tends to be a few irrelevant documents in the feedback set. Due to this the performance and stability of standard PRF techniques is adversely affected.

For example in Table 10.1, we see the huge performance increase possible in the Model-Based Feedback technique if the irrelevant documents in the top 10 are identified. We see there is a possible 10-25% increase in performance if the irrelevant documents in the feedback set are identfied, and even more astonishing 30-65% increase if PRF is done using the top $k$ relevant documents.

| **ID** | **LM** | **MF** | **MF on Rel. Docs. in top $k$** | | **MF on Top $k$ Rel. Docs** | |
|---|---|---|---|---|---|---|
| | MAP | MAP | MAP | %Change from MF | MAP | %Change from MF |
| LAT | 0.434 | 0.442 | 0.56 | **26.7%** | 0.67 | **51.7%** |
| GH | 0.382 | 0.405 | 0.486 | **20%** | 0.603 | **48.8%** |
| AP | 0.277 | 0.327 | 0.37 | **13.2%** | 0.454 | **38.7%** |
| WSJ | 0.266 | 0.303 | 0.337 | **11.1%** | 0.394 | **30.2%** |
| SJM | 0.207 | 0.235 | 0.287 | **22%** | 0.388 | **65.1%** |

Table 10.1: Effect of Irrelevant Docs. in top $k$ ($k = 10$)

Thus we see that a standard PRF technique such as Model-Based Feedback itself, benefits so largely by the identification of these irrelevant documents and further benefits by replacing the irrelevant documents by relevant documents. Since there has been no previous work in this area,

we are motivated to try and find a technique to identify the irrelevant documents in the feedback set.

## 10.2 Difficulties Faced

In this section we will try to understand why this task so so hard, and what could be the possible reasons. Some of the common reasons are;

1. *Synonymy*: This is a very difficult problem to solve. This occurs when the query contains a particular term, but the documents in the corpus(including most of the relevant documents) contains the synonym of the word. Thus thought the query may contain the term "cougar", if the documents in the collection use the phrase "mountain lion" instead, then it is very difficult to separate the relevant and irrelevant documents. One solution is to use a lexical resource such as a thesaurus, but this would require these additional resources.

2. *Ambiguous Query*: This is another very difficult problem to solve. If the query itself is inherently ambiguous, then it is again very difficult to understand the intent of the user, and hence identifying the irrelevant documents in the top $k$. An example of such a query would be "Mediterranean Olive Oil". Here the query could refer to production of olive oil in the Mediterranean countries, or could refer to use of Olive Oil in Mediterranean cooking. Note sometimes this problem could arise due removal of stopwords and stemming of the query.

3. *Topic Only*: Since we use only the topics of the query, in some cases the complete intent is not conveyed. In many queries while the description and narration of the query convey the complete intent of the user, the topic may not do so. This is particularly the case when a specific piece of information is required. For example if the query topic is "American troops in Iraq", but the description is "Find the expenditure incurred by America in sending troops to Iraq". In this case though a document may talk about the American troops in Iraq, and may thus seem very similar to other relevant documents, if it misses the monetary aspect it is irrelevant. Hence in such cases it is almost impossible to identify such irrelevant documents.

4. *Spelling/Transcoding Errors*: This is particularly a problem for monolingual retrieval in other languages. In such cases, there tend to be mistakes in spelling, or transcoding errors when converting a name from English to that language. Hence due to these spelling mistakes, it again becomes difficult to identify the irrelevant documents from the relevant documents.

5. *Vaguely Related Topic*: One of the biggest challenges faced in this task is due to the presence of documents which talk about a vaguely related topic, but have large number of occurences of the query terms, since the topics share those terms in common. This is a very difficult problem to tackle, as by using the query terms alone it is very detect such documents. However, in general these documents use the query terms not in their most significant parts. All in all, such documents are almost impossible to detect. What makes the situation worse is that such documents are prevalent in the feedback sets of most queries, in the test collections. Hence the performance of any system which tries to identify these irrelevant documents, is going to be severely affected by these documents. Thus this effectively forces an upper bound on the performance of any such system.

Table 10.2 gives us the total number of relevant and irrelevant documents in the top 10, for different collections. The last column gives the ratio of irrelevant documents to relevant documents.

| ID | Total Number of Relevant | Total Number of Irrelevant | Ratio |
|---|---|---|---|
| LAT | 467 | 753 | 1.61 |
| GH | 560 | 960 | 1.71 |
| AP | 663 | 827 | 1.25 |
| WSJ | 686 | 814 | 1.19 |
| SJM | 286 | 654 | 2.29 |

Table 10.2: Number of Relevant and Irrelevant Documents in the PRF set

Thus this gives us an idea of the average composition of the feedback set, in different collections.

## 10.3 Possible Distinguishing Properties

Although this is a difficult task, there are some possible tools which can help us in the task. These are some features/resources which if used can help distinguish some of the relevant documents from others.

1. *Number of Query Words*: The number of query words a document contains, can be a good indicator of its' relevance/irrelevance. In particular if a document is missing a query term then it is more likely to be irrelevant. This is especially true when the query term which is missing is the highest IDF query term. In table 10.3, we see the number of such cases in the LAT and AP collections. From the ratio of these number between relevant and irrelevant, we see that indeed is true that a document having a query term missing, is more likely to be irrelevant.

2. *Total IDF of Query Words Present*: Another potential distinguishing property of irrelevant and relevant documents, is the total fraction of IDF missing from the document. For example if the query has 3 terms $q_1, q_2, q_3$ with IDFs $i_1, i_2, i_3$ respectively, then if a document is missing the term $q_2$ the fraction of IDF missing is $\frac{i_2}{i_1+i_2+i_3}$. This is another useful indicator, because even if a relevant document is missing a query term, it could be a less important term, whereas an irrelevant document is likely to be missing any of them. Since IDF is a good approximation of importance, the fraction of IDF missing could be a useful property.Some statistics to show the same are there in table 10.3.

3. *Proximity of Query Words*: Another potential property is the proximity of the query terms in a document. Since a query tends to be a set of terms which are related in the context of a specific concept, relevant documents tend to have the query terms close to each other. However irrelevant documents tend to have them more spread out. This was verified on the LAT and AP corpora. Zhai and Tao Tao [TZ07] had studied some proximity measures in the context of information retrieval, and suggested some measures, which performed well. Thus using one of these proximity based measures, can also help in identifying irrelevant documents

4. *Synonyms of Query Words*: Another possible property to help in identifying irrelevant documents from relevant documents, is the presence of synonyms of query terms. However this requires a resource such as a thesaurus or WordNet.

5. *Distance from MLE Feedback Distribution*: The distance of a document from the MLE distribution of the feedback set, is another property which can be used to identify irrelevant documents. The concept behind this is: When there are a majority of relevant documents in the top $k$ then the MLE distribution tends to be close to them. Hence in general an irrelevant document in the top $k$ will lie quite far from this distribution. However if there were a majority of irrelevant documents in the top $k$, it generally is the case that they are not of the same topic, and hence the MLE distribution is not skewed, due to which a relevant document in the top $k$ will not be too far from the MLE distribution. Thus in this case the distance from the MLE distribution is not very useful property. Hence this property will be useful when there are a significant number of relevant documents in the top $k$.

6. *Distance from MLE Pseudo-Irrelevant Distribution*: Another potentially powerful property which we can use to identify these irrelevant documents, is the distance of a document from the MLE distribution of the pseudo-irrelevant set as obtained in chapter 8. This is because the pseudo-irrelevant distribution contains a majority of irrelevant documents and hence the irrelevant documents in the top $k$ are more likely to be closer to the distribution. We verified this experimentally as well. On the LAT corpus, the average KL-Divergence score of a relevant document in the top $k$ from the pseudo-irrelevant distribution was found to be 0.654, while the corresponding figure for an irrelevant document in the top $k$ was found to be 0.787; thus confirming that the irrelevant documents in the top $k$ tend to be closer to the pseudo-irrelevant distribution than the relevant documents in the top $k$.

| **Feature** | **LAT** | | **AP** | |
|---|---|---|---|---|
| | Rel | Irrel | Rel | Irrel |
| No. of Docs with Query Term Missing | 71 | 312 | 247 | 469 |
| No. of Docs with Highest IDF Query Term Missing | 14 | 81 | 36 | 84 |
| Average Fraction of IDF missing | 0.306 | 0.344 | 0.263 | 0.29 |
| No. of Docs with 50% of Max. Query Term IDF Missing | 5 | 46 | 14 | 43 |

Table 10.3: Some statistics of Relevant And Irrelevant Documents

## 10.4 High-Precision Irrelevancy Classifier

All of the above mentioned properties can help us in identifying irrelevant documents. However individually using them would be heuristic, and not a effective. Hence our idea is to learn a high-precision irrelevancy classifier with the above mentioned properties used as features. Thus for each document we would compute the feature values for each of these features, and then learn the feature weights to help us make a decision.

| Features Used | Description |
|---|---|
| $\|Q\| - \sum_{q_i \in D} 1$ | Number of Query Terms Missing in the Document |
| $\frac{\sum_{q_i \in D} Count(q_i)}{\|Q\|}$ | Average Number Of Occurrences of Query Terms |
| $1 - \frac{\sum_{q_i \in D} IDF(q_i)}{\sum_{q_i \in Q} IDF(q_i)}$ | Fraction of IDF missing |
| $-KLD(\theta_{PRF}^{MLE}, \theta_D)$ | Negative of KL-Divergence between the MLE Distribution of PRF set and Document Model |
| $-KLD(\theta_{PIR}^{MLE}, \theta_D)$ | Negative of KL-Divergence between the MLE Distribution of Pseudo-Irrelevant set and Document Model |

Table 10.4: Description of Features Used in the Implementation

The key idea here is that each of the above features may not be useful on their own. Hence they need to be combined in an appropriate manner. At the same time, the classifier has to be of high-accuracy, as wrongly classifying a relevant document can lead to a large penalty. Thus in our method we will use a training set of queries, and learn the required feature weights, as well as find a margin for high-precision classification on a validation set, and then finally classify the feedback sets of the remaining test queries.

**Algorithm 4** : HighPrecisionIrrelevancyClassifier

1: Compute the feature values for the top $k$ documents of all the queries
2: Use the feature values and given labels to learn a weight vector on the training set
3: Use the training and validation sets, to tune the classification margin for required precision value.
4: Run on the test set and Identify irrelevant documents

This is the algorithm we propose for identifying irrelevant documents. Currently we are in the process of implementing our method, and hence do not have any results yet.

## 10.5 Experimental Setup

In this section, we discuss the setup used in our experiments where we built such a high-performance irrelevance classifier. As before we use the CLEF and TREC collections for our experiments. We perform the same pre-processing as before, and use the same baselines and initial retrieval algorithms. As before, we only consider those topics which have atleast 1 relevant document. In all of our experiment, we set $k$ as 10, *i.e.* the size of the feedback set is uniformly set as 10. The features used are given in Table 10.4.

This is a relatively simple classifier as we only use 5 features, and do not have any features corresponding to Synonyms, Proximity Measures of Query Terms or Variance of Query Term Counts. We can only expect perform to improve on adding such features. Intuitively we expect these feature to take the following values:

1. *Missing Query Terms*: The larger this value, the more likely it is to be irrelevant. Hence we expect the corresponding feature weight to be negative.
2. *Average Query Term Count*: The larger this value, the more likely it is to be relevant. Hence we expect the corresponding feature weight to be positive.
3. *Missing IDF Fraction*: The larger this value, the more likely it is to be irrelevant. Hence we expect the corresponding feature weight to be negative.
4. *Negative of Distance to MLE PRF Distribution*: The larger this value, the more likely it is to be relevant. Hence we expect the corresponding feature weight to be positive.
5. *Negative of Distance to MLE PIR Distribution*: The larger this value, the more likely it is to be irrelevant. Hence we expect the corresponding feature weight to be negative.

Note that we do not any feature normalization or other operators on the feature values. As before, we use a logistic classifier to learn the feature weights, while providing it +ve (*relevant documents*) and -ve (*irrelevant documents*) as examples. For this purpose we use LibLinear as before due to its' ease of availability. We then also ran experiment, where we used the scores of the classifier, and eliminated documents from the feedback set before performing PRF.

## 10.6 Results

One of the first things we observed on running the logistic classifier, is that the signs of the weights corresponds to the intuition. On further experimentation, we found Features 1 and 5 to be the most useful, while Features 3 and 4 just behind them, with Feature 5 being the least useful. In fact, the performance of the classifier thus does not change much on removing Feature 2. Hence we realized it is necessary to also add Variance of Query Term Counts as a feature, as the mean alone is not informative enough, since it does not tell you if all query terms occur frequently in the documents, or if there was one query term which occurs very frequently in that document.

The training and test set performances of the classifier learnt are given in Tables 10.5 and 10.6 respectively. In this tables we report performances observed on varying the percentage of data used as training data. The entries in the Tables are of the form:

"$x\%$: *Val*$\leq y$; $n_{ir}$ *Irrel* + $n_r$ *Rel*"

This can be interpreted as: the lowest $x\%$ of documents are composed of $n_r$ relevant documents and $n_{ir}$ irrelevant documents. Ideally, we would like the $n_{ir}$ values to be large and $n_r$ values to

| Coll. | 30% Training Data | 40% Training Data | 50% Training Data |
|---|---|---|---|
| LA94 | 5%: Val<= -0.837: 12 Irrel + 6 Rel | 5%: Val<= -0.994: 19 Irrel + 5 Rel | 5%: Val<= -0.936: 22 Irrel + 8 Rel |
| | 10%: Val<= -0.803: 26 Irrel + 10 Rel | 10%: Val<= -0.899: 38 Irrel + 10 Rel | 10%: Val<= -0.85: 49 Irrel + 12 Rel |
| | 20%: Val<= -0.704: 54 Irrel + 18 Rel | 20%: Val<= -0.702: 77 Irrel + 19 Rel | 20%: Val<= -0.698: 101 Irrel + 21 Rel |
| | 25%: Val<= -0.539: 70 Irrel + 20 Rel | 25%: Val<= -0.454: 98 Irrel + 22 Rel | 25%: Val<= -0.544: 122 Irrel + 30 Rel |
| | 30%: Val<= -0.04: 85 Irrel + 23 Rel | 30%: Val<= 0.061: 117 Irrel + 27 Rel | 30%: Val<= 0.142: 146 Irrel + 37 Rel |
| | 40%: Val<= -0.016: 108 Irrel + 36 Rel | 40%: Val<= 0.133: 152 Irrel + 40 Rel | 40%: Val<= 0.221: 187 Irrel + 57 Rel |
| | 50%: Val<= 0.0060: 137 Irrel + 43 Rel | 50%: Val<= 0.212: 183 Irrel + 57 Rel | 50%: Val<= 0.278: 229 Irrel + 76 Rel |
| | 75%: Val<= 0.08: 189 Irrel + 81 Rel | 75%: Val<= 0.388: 250 Irrel + 110 Rel | 75%: Val<= 0.431: 313 Irrel + 144 Rel |
| | 100%: Val<= 1.036: 228 Irrel + 132 Rel | 100%: Val<= 2.102: 301 Irrel + 179 Rel | 100%: Val<= 1.611: 378 Irrel + 232 Rel |
| GH95 | 5%: Val<= -1.474: 22 Irrel + 0 Rel | 5%: Val<= -1.374: 30 Irrel + 0 Rel | 5%: Val<= -1.366: 38 Irrel + 0 Rel |
| | 10%: Val<= -1.356: 45 Irrel + 0 Rel | 10%: Val<= -1.218: 57 Irrel + 3 Rel | 10%: Val<= -1.224: 73 Irrel + 3 Rel |
| | 20%: Val<= -1.201: 82 Irrel + 8 Rel | 20%: Val<= -1.096: 102 Irrel + 18 Rel | 20%: Val<= -1.051: 134 Irrel + 18 Rel |
| | 25%: Val<= -1.162: 97 Irrel + 15 Rel | 25%: Val<= -1.034: 131 Irrel + 19 Rel | 25%: Val<= -0.964: 168 Irrel + 22 Rel |
| | 30%: Val<= -1.113: 119 Irrel + 16 Rel | 30%: Val<= -0.956: 157 Irrel + 23 Rel | 30%: Val<= -0.741: 201 Irrel + 27 Rel |
| | 40%: Val<= -0.756: 156 Irrel + 24 Rel | 40%: Val<= 0.142: 203 Irrel + 37 Rel | 40%: Val<= 0.226: 255 Irrel + 49 Rel |
| | 50%: Val<= 0.24: 192 Irrel + 33 Rel | 50%: Val<= 0.225: 240 Irrel + 60 Rel | 50%: Val<= 0.32: 310 Irrel + 70 Rel |
| | 75%: Val<= 0.478: 257 Irrel + 80 Rel | 75%: Val<= 0.446: 331 Irrel + 119 Rel | 75%: Val<= 0.659: 423 Irrel + 147 Rel |
| | 100%: Val<= 3.13: 307 Irrel + 143 Rel | 100%: Val<= 3.16: 398 Irrel + 202 Rel | 100%: Val<= 4.143: 495 Irrel + 265 Rel |
| AP | 5%: Val<= -1.784: 21 Irrel + 1 Rel | 5%: Val<= -1.314: 27 Irrel + 2 Rel | 5%: Val<= -0.778: 31 Irrel + 6 Rel |
| | 10%: Val<= -1.58: 39 Irrel + 5 Rel | 10%: Val<= -1.087: 52 Irrel + 7 Rel | 10%: Val<= -0.583: 58 Irrel + 16 Rel |
| | 20%: Val<= -0.867: 78 Irrel + 10 Rel | 20%: Val<= -0.588: 99 Irrel + 19 Rel | 20%: Val<= -0.293: 116 Irrel + 32 Rel |
| | 25%: Val<= -0.759: 95 Irrel + 15 Rel | 25%: Val<= -0.528: 116 Irrel + 31 Rel | 25%: Val<= -0.244: 141 Irrel + 44 Rel |
| | 30%: Val<= -0.654: 109 Irrel + 23 Rel | 30%: Val<= -0.471: 135 Irrel + 42 Rel | 30%: Val<= -0.18: 163 Irrel + 59 Rel |
| | 40%: Val<= -0.025: 140 Irrel + 36 Rel | 40%: Val<= -0.049: 172 Irrel + 64 Rel | 40%: Val<= 0.087: 207 Irrel + 89 Rel |
| | 50%: Val<= 0.159: 172 Irrel + 48 Rel | 50%: Val<= 0.126: 213 Irrel + 82 Rel | 50%: Val<= 0.2: 262 Irrel + 108 Rel |
| | 75%: Val<= 0.582: 222 Irrel + 108 Rel | 75%: Val<= 0.423: 287 Irrel + 155 Rel | 75%: Val<= 0.425: 362 Irrel + 193 Rel |
| | 100%: Val<= 2.786: 254 Irrel + 186 Rel | 100%: Val<= 1.897: 342 Irrel + 248 Rel | 100%: Val<= 1.499: 439 Irrel + 301 Rel |
| WSJ | 5%: Val<= -0.575: 18 Irrel + 4 Rel | 5%: Val<= -0.437: 26 Irrel + 4 Rel | 5%: Val<= -0.353: 31 Irrel + 6 Rel |
| | 10%: Val<= -0.41: 41 Irrel + 4 Rel | 10%: Val<= -0.258: 49 Irrel + 11 Rel | 10%: Val<= -0.191: 59 Irrel + 16 Rel |
| | 20%: Val<= -0.176: 72 Irrel + 18 Rel | 20%: Val<= -0.054: 92 Irrel + 28 Rel | 20%: Val<= -0.027: 110 Irrel + 40 Rel |
| | 25%: Val<= -0.084: 84 Irrel + 28 Rel | 25%: Val<= -0.0050: 115 Irrel + 35 Rel | 25%: Val<= 0.048: 133 Irrel + 54 Rel |
| | 30%: Val<= -0.043: 105 Irrel + 30 Rel | 30%: Val<= 0.027: 133 Irrel + 47 Rel | 30%: Val<= 0.099: 161 Irrel + 64 Rel |
| | 40%: Val<= 0.043: 134 Irrel + 46 Rel | 40%: Val<= 0.128: 176 Irrel + 64 Rel | 40%: Val<= 0.172: 210 Irrel + 90 Rel |
| | 50%: Val<= 0.183: 166 Irrel + 59 Rel | 50%: Val<= 0.208: 209 Irrel + 91 Rel | 50%: Val<= 0.228: 249 Irrel + 126 Rel |
| | 75%: Val<= 0.576: 216 Irrel + 121 Rel | 75%: Val<= 0.458: 284 Irrel + 166 Rel | 75%: Val<= 0.388: 345 Irrel + 217 Rel |
| | 100%: Val<= 2.558: 254 Irrel + 196 Rel | 100%: Val<= 1.674: 354 Irrel + 246 Rel | 100%: Val<= 1.154: 438 Irrel + 312 Rel |
| SJM | 5%: Val<= -1.444: 14 Irrel + 0 Rel | 5%: Val<= -2.49: 18 Irrel + 0 Rel | 5%: Val<= -1.76: 23 Irrel + 0 Rel |
| | 10%: Val<= -1.242: 27 Irrel + 1 Rel | 10%: Val<= -1.695: 37 Irrel + 0 Rel | 10%: Val<= -1.284: 46 Irrel + 1 Rel |
| | 20%: Val<= -0.77: 52 Irrel + 4 Rel | 20%: Val<= -0.934: 67 Irrel + 7 Rel | 20%: Val<= -0.704: 86 Irrel + 8 Rel |
| | 25%: Val<= -0.71: 63 Irrel + 7 Rel | 25%: Val<= -0.862: 85 Irrel + 7 Rel | 25%: Val<= -0.614: 109 Irrel + 8 Rel |
| | 30%: Val<= -0.498: 76 Irrel + 8 Rel | 30%: Val<= -0.8: 101 Irrel + 10 Rel | 30%: Val<= -0.567: 126 Irrel + 15 Rel |
| | 40%: Val<= -0.088: 97 Irrel + 15 Rel | 40%: Val<= -0.432: 134 Irrel + 14 Rel | 40%: Val<= -0.327: 166 Irrel + 22 Rel |
| | 50%: Val<= -0.0030: 120 Irrel + 20 Rel | 50%: Val<= 0.058: 159 Irrel + 26 Rel | 50%: Val<= 0.111: 198 Irrel + 37 Rel |
| | 75%: Val<= 0.282: 165 Irrel + 45 Rel | 75%: Val<= 0.285: 216 Irrel + 61 Rel | 75%: Val<= 0.401: 273 Irrel + 79 Rel |
| | 100%: Val<= 2.095: 204 Irrel + 76 Rel | 100%: Val<= 1.83: 268 Irrel + 102 Rel | 100%: Val<= 2.226: 337 Irrel + 133 Rel |

Table 10.5: Performance of the Classifier on the Given Training Sets.

| Coll. | 30% Training Data | 40% Training Data | 50% Training Data |
|---|---|---|---|
| LA94 | 5%: Val<= -0.869: 37 Irrel + 6 Rel | 5%: Val<= -1.01: 33 Irrel + 4 Rel | 5%: Val<= -1.013: 27 Irrel + 3 Rel |
| | 10%: Val<= -0.85: 78 Irrel + 8 Rel | 10%: Val<= -0.971: 64 Irrel + 10 Rel | 10%: Val<= -0.945: 55 Irrel + 6 Rel |
| | 20%: Val<= -0.81: 151 Irrel + 21 Rel | 20%: Val<= -0.872: 129 Irrel + 19 Rel | 20%: Val<= -0.839: 108 Irrel + 14 Rel |
| | 25%: Val<= -0.767: 184 Irrel + 31 Rel | 25%: Val<= -0.776: 155 Irrel + 30 Rel | 25%: Val<= -0.78: 131 Irrel + 21 Rel |
| | 30%: Val<= -0.654: 216 Irrel + 42 Rel | 30%: Val<= -0.591: 184 Irrel + 38 Rel | 30%: Val<= -0.701: 153 Irrel + 30 Rel |
| | 40%: Val<= -0.029: 270 Irrel + 74 Rel | 40%: Val<= 0.108: 232 Irrel + 64 Rel | 40%: Val<= 0.206: 196 Irrel + 48 Rel |
| | 50%: Val<= -0.0040: 323 Irrel + 107 Rel | 50%: Val<= 0.158: 274 Irrel + 96 Rel | 50%: Val<= 0.258: 236 Irrel + 69 Rel |
| | 75%: Val<= 0.096: 432 Irrel + 213 Rel | 75%: Val<= 0.335: 369 Irrel + 186 Rel | 75%: Val<= 0.37: 308 Irrel + 149 Rel |
| | 100%: Val<= 0.748: 525 Irrel + 335 Rel | 100%: Val<= 1.601: 452 Irrel + 288 Rel | 100%: Val<= 1.286: 375 Irrel + 235 Rel |
| GH95 | 5%: Val<= -1.274: 49 Irrel + 4 Rel | 5%: Val<= -1.22: 45 Irrel + 1 Rel | 5%: Val<= -1.172: 33 Irrel + 5 Rel |
| | 10%: Val<= -1.12: 89 Irrel + 18 Rel | 10%: Val<= -1.037: 77 Irrel + 15 Rel | 10%: Val<= -0.881: 58 Irrel + 18 Rel |
| | 20%: Val<= 0.2: 164 Irrel + 50 Rel | 20%: Val<= 0.19: 144 Irrel + 40 Rel | 20%: Val<= 0.256: 109 Irrel + 43 Rel |
| | 25%: Val<= 0.243: 200 Irrel + 67 Rel | 25%: Val<= 0.233: 173 Irrel + 57 Rel | 25%: Val<= 0.326: 130 Irrel + 60 Rel |
| | 30%: Val<= 0.28: 231 Irrel + 90 Rel | 30%: Val<= 0.285: 202 Irrel + 74 Rel | 30%: Val<= 0.39: 153 Irrel + 75 Rel |
| | 40%: Val<= 0.356: 291 Irrel + 137 Rel | 40%: Val<= 0.359: 252 Irrel + 116 Rel | 40%: Val<= 0.493: 192 Irrel + 112 Rel |
| | 50%: Val<= 0.427: 352 Irrel + 183 Rel | 50%: Val<= 0.417: 301 Irrel + 159 Rel | 50%: Val<= 0.588: 231 Irrel + 149 Rel |
| | 75%: Val<= 0.703: 502 Irrel + 300 Rel | 75%: Val<= 0.721: 425 Irrel + 265 Rel | 75%: Val<= 0.986: 340 Irrel + 230 Rel |
| | 100%: Val<= 2.86: 653 Irrel + 417 Rel | 100%: Val<= 2.877: 562 Irrel + 358 Rel | 100%: Val<= 3.805: 465 Irrel + 295 Rel |
| AP | 5%: Val<= -2.333: 38 Irrel + 14 Rel | 5%: Val<= -1.801: 33 Irrel + 12 Rel | 5%: Val<= -1.252: 26 Irrel + 11 Rel |
| | 10%: Val<= -1.564: 74 Irrel + 31 Rel | 10%: Val<= -1.259: 62 Irrel + 28 Rel | 10%: Val<= -0.816: 54 Irrel + 21 Rel |
| | 20%: Val<= -0.88: 139 Irrel + 71 Rel | 20%: Val<= -0.787: 119 Irrel + 61 Rel | 20%: Val<= -0.496: 106 Irrel + 44 Rel |
| | 25%: Val<= -0.771: 170 Irrel + 92 Rel | 25%: Val<= -0.65: 150 Irrel + 75 Rel | 25%: Val<= -0.361: 128 Irrel + 59 Rel |
| | 30%: Val<= -0.688: 206 Irrel + 109 Rel | 30%: Val<= -0.567: 174 Irrel + 96 Rel | 30%: Val<= -0.32: 151 Irrel + 74 Rel |
| | 40%: Val<= -0.49: 269 Irrel + 151 Rel | 40%: Val<= -0.391: 238 Irrel + 122 Rel | 40%: Val<= -0.193: 205 Irrel + 95 Rel |
| | 50%: Val<= -0.021: 319 Irrel + 206 Rel | 50%: Val<= -0.104: 283 Irrel + 167 Rel | 50%: Val<= -0.036: 242 Irrel + 133 Rel |
| | 75%: Val<= 0.45: 465 Irrel + 322 Rel | 75%: Val<= 0.305: 405 Irrel + 270 Rel | 75%: Val<= 0.309: 333 Irrel + 229 Rel |
| | 100%: Val<= 2.732: 573 Irrel + 477 Rel | 100%: Val<= 1.899: 485 Irrel + 415 Rel | 100%: Val<= 1.53: 388 Irrel + 362 Rel |
| WSJ | 5%: Val<= -0.539: 41 Irrel + 11 Rel | 5%: Val<= -0.448: 38 Irrel + 7 Rel | 5%: Val<= -0.465: 33 Irrel + 4 Rel |
| | 10%: Val<= -0.355: 76 Irrel + 29 Rel | 10%: Val<= -0.308: 68 Irrel + 22 Rel | 10%: Val<= -0.326: 56 Irrel + 19 Rel |
| | 20%: Val<= -0.129: 137 Irrel + 73 Rel | 20%: Val<= -0.137: 119 Irrel + 61 Rel | 20%: Val<= -0.163: 103 Irrel + 47 Rel |
| | 25%: Val<= -0.047: 164 Irrel + 98 Rel | 25%: Val<= -0.089: 146 Irrel + 79 Rel | 25%: Val<= -0.114: 126 Irrel + 61 Rel |
| | 30%: Val<= 0.028: 192 Irrel + 123 Rel | 30%: Val<= -0.015: 165 Irrel + 105 Rel | 30%: Val<= -0.062: 144 Irrel + 81 Rel |
| | 40%: Val<= 0.154: 247 Irrel + 173 Rel | 40%: Val<= 0.089: 214 Irrel + 146 Rel | 40%: Val<= 0.058: 182 Irrel + 118 Rel |
| | 50%: Val<= 0.269: 303 Irrel + 222 Rel | 50%: Val<= 0.179: 260 Irrel + 190 Rel | 50%: Val<= 0.145: 223 Irrel + 152 Rel |
| | 75%: Val<= 0.627: 426 Irrel + 361 Rel | 75%: Val<= 0.413: 364 Irrel + 311 Rel | 75%: Val<= 0.332: 305 Irrel + 257 Rel |
| | 100%: Val<= 2.415: 560 Irrel + 490 Rel | 100%: Val<= 1.626: 460 Irrel + 440 Rel | 100%: Val<= 1.056: 376 Irrel + 374 Rel |
| SJM | 5%: Val<= -1.873: 27 Irrel + 6 Rel | 5%: Val<= -1.919: 19 Irrel + 9 Rel | 5%: Val<= -1.404: 13 Irrel + 10 Rel |
| | 10%: Val<= -1.327: 51 Irrel + 15 Rel | 10%: Val<= -1.609: 40 Irrel + 17 Rel | 10%: Val<= -1.188: 28 Irrel + 19 Rel |
| | 20%: Val<= -0.907: 97 Irrel + 35 Rel | 20%: Val<= -0.927: 78 Irrel + 36 Rel | 20%: Val<= -0.699: 61 Irrel + 33 Rel |
| | 25%: Val<= -0.767: 122 Irrel + 43 Rel | 25%: Val<= -0.854: 97 Irrel + 45 Rel | 25%: Val<= -0.599: 74 Irrel + 43 Rel |
| | 30%: Val<= -0.694: 144 Irrel + 54 Rel | 30%: Val<= -0.787: 116 Irrel + 55 Rel | 30%: Val<= -0.518: 89 Irrel + 52 Rel |
| | 40%: Val<= -0.576: 191 Irrel + 73 Rel | 40%: Val<= -0.671: 155 Irrel + 73 Rel | 40%: Val<= -0.391: 120 Irrel + 68 Rel |
| | 50%: Val<= -0.35: 236 Irrel + 94 Rel | 50%: Val<= -0.502: 193 Irrel + 92 Rel | 50%: Val<= -0.113: 155 Irrel + 80 Rel |
| | 75%: Val<= 0.165: 342 Irrel + 153 Rel | 75%: Val<= 0.231: 287 Irrel + 140 Rel | 75%: Val<= 0.361: 231 Irrel + 121 Rel |
| | 100%: Val<= 1.76: 450 Irrel + 210 Rel | 100%: Val<= 1.47: 386 Irrel + 184 Rel | 100%: Val<= 1.754: 317 Irrel + 153 Rel |

Table 10.6: Performance of the Classifier on the Test Sets.

| % of Feedback Set Eliminated | Training Set MAP | Test Set MAP |
|---|---|---|
| 0% (i.e. MBF) | 0.5014 | 0.3862 |
| 20% | 0.5014 | 0.3879 |
| 30% | 0.4993 | 0.3901 |
| 40% | 0.4972 | 0.3974 |

Table 10.7: LA94 Dataset: MAP Value performance of the Elimination Algorithm, and how it varies with number of documents eliminated

be small. Thus as we can see the classifier is able to do a good job in distinguishing relevant documents from irrelevant documents. For example the Test Set performance with 30% Training Data on the LA94 collection, has the lowest 20% composed of 151 Irrelevant documents and 21 Relevant Documents. This is a much larger ratio than that of the Number of Irrelevant Documents in the Test Set to the Number of Relevant Documents which is 525:335.

Hence the classifier's performance indicates that this approach of identifying irrelevant documents is a very potent idea, as even with just 5 simple features and using a logistic classifier we are able to identify irrelevant documents quite accurately. Another interesting observation is that the performance on the training data and the test data are not similar, and hence we conclude that the classifier is overfitting, Hence we expect that changing the regularizer should result in better test set performance while sacrificing on the training set performance. We also observe that the test set performance is quite steady, even when the percentage of training set data decreases. Hence we can hope that the classifier will be able to perform well even with small amounts of training data. One aspect which is missing in our experiments is that of $k$-fold validation, as in our case, we fixed the training data and test data. A better choice would have been to use $k$-fold validation and report results.

| % of Feedback Set Eliminated | Training Set MAP | Test Set MAP |
|---|---|---|
| 0% (i.e. MBF) | 0.2485 | 0.2272 |
| 20% | 0.2421 | 0.2278 |
| 30% | 0.2323 | 0.2276 |
| 40% | 0.2317 | 0.228 |

Table 10.8: SJM Dataset: MAP Value performance of the Elimination Algorithm, and how it varies with number of documents eliminated

### 10.6.1 Using the Classifier Scores: Elimination Algorithm

In this subsection, we explore a method of using the scores of the above-mentioned classifier. In our approach, we use a threshold value, which depends on the % of documents we want to eliminate. For every query, we first run the initial retrieval algorithm and obtain the top 10 documents. We then run the classifier on these documents and obtain the scores of these documents. We then eliminate all those documents from the feedback set, whose score is below the threshold value. We then use the remaining documents in the feedback set, to perform PRF as per the MBF algorithm.

Tables 10.7 and 10.8 show the performance of this simplistic method on the LA Times and San Jose Mercury Datasets, and how the MAP value performance varies based on this threshold. One key point to note here is that, we observed that MBF performs worse when the number of feedback document is reduced from 10. However we note that we perform almost as well as MBF on all 10 documents, even when we remove 2/3/4 documents on average from the feedback set. Even better, we notice improvements on the training data performance as seen in Table 10.7. This is due to the better accuracy of classification on these datasets. Hence, we can hope that a better classifier will lead to larger MAP values. In this case too, a $k$-fold validation would have been the optimal experiment to perform rather than fix the datasets.

## 10.7 Extending this Model

As we noted, the Elimination Algorithm proposed in the previous section, was too simplistic, and simultaneously rather drastic, since it reduced the feedback set size. However we noted, that even this simple algorithm was able to perform competitively as compared to the baseline MBF performance. Thus we are motivated to improve this algorithm, in the hope of doing better. In this section, we review several possible improvements possible to the model and the classifier. Some improvements possible in the classifier are:

- Use more features such as the variance of query terms. Also avoid overfitting.
- Incorporate presence of synonyms and morphological variants as features.
- Incorporate proximity of query terms as a feature.
- Using an SVM-based classifier. (May not lead to improvements though)
- Decision Tree-based Classifier: Since decision trees work by breaking up the decision space into squares/rectangles and finding good classifiers for each of these smaller units, such an approach in this situation may lead to improvements, as we suspect there to be an implicit, hidden division of the space into such kinds of rectangles.
- Cost-Based Classification: Here while we equally penalize both kinds of misclassifications, in reality since we want a high-precision irrelevance classifier, we want to penalize the classification of relevant documents as irrelevant much more, since in this case we end up losing such documents. These kinds of approaches have successfully been tried in biological applications, where predicting as someone who is suspected of having cancer, as no threat versus the opposite are very different scenarios. Hence we can expect that using such a classifier, should lead to large improvements.

Similarly we also have ways of improving the Elimination Algorithm proposed in the previous subsection. Some of the possible improvements are:

- Document Weighting Scheme: Instead of completely removing a document from the feedback set, and thus reducing the size of the feedback set, we can alternately use the scores of the classifier to learn weights (representing relative importance of the documents) which can then be used in a document-weighting framework. We will see more of this idea in the next chapter.
- Other Methods of Negative Feedback: We can use the predicted irrelevant documents as negative feedback, and use methods similar to those discussed in Chapter 9 to improve PRF.

## 10.8 Further Interesting Questions

There are some interesting questions which arise based on our observations. Some of them are:

1. *Identifying Irrelevant documents outside the top k*: Another question which arises if this can be used to identify irrelevant documents outside the top $k$. Alternately, we could ask if this method can be generalized to identifying high-scoring irrelevant documents, and not just irrelevant documents in the top $k$.

2. *High-Precision Relevance Classifier*: Analogous to learning an irrelevance classifier, we are also trying to learn a high-precision relevance classifier. Note that this differs from the irrelevance classifier, in the weights learnt and threshold used for classification. We refer to this concept later, when we talk about document resampling and document weighting.

# Chapter 11

# Fuzzy Relevance and Document Weighting

## 11.1 Motivation

As seem in chapter 10, there are many inherent complexities involved in identifying irrelevant documents. Though we have proposed a method to identify irrelevant documents, due to it being a high-precision method, the recall levels are brought down due to this. Hence there will still exist some relevant documents in the top $k$, which are difficult to identify. However we are not completely helpless in these cases as well, as the irrelevance classifier and feature values do give us some information as to the extent of the document's relevance. Hence we should put this information to use as well, and not treat all the residual documents in the feedback set similarly.

PRF methods like Model-Based Feedback treat all the documents in the feedback set identically. However this is not optimal especially when some knowledge regarding the extent of a document's relevance is given. Thus if a document $X$ is expected to be relevant with a high degree of confidence, while document $Y$ is expected to be irrelevant, then treating them identically in a PRF method does not make sense. Hence we propose the concept of fuzzy relevance and document weighting.

## 11.2 Fuzzy Relevance

In chapter 10 we proposed a method of building a high-precision irrelevance classifier, while in section 10.8 we proposed building an analogous high-precision relevance classifier. Now using these two classifiers we can obtain the classification scores for the documents in the residual feedback set(Feedback set got after removing the irrelevant documents flagged by the irrelevance classifier). These scores give us some indication of the document's chances of being relevant. Hence we propose the concept of **fuzzy relevance** to refer to the case where we are unsure of a document being relevant or irrelevant, but have some information regarding its' chances of being relevant/irrelevant. Thus the fuzzy relevance of a document $d_i$ is defined as $F_{d_i}$, and is defined as a function of the

documents' score $R_{d_i}$ as per the relevance classifier's score of the document, and analogously $I_{d_i}$ referring to the score of the irrelevance classifier. In other words $F_{d_i} = SomeFuncn(R_{d_i}, I_{d_i})$, where $F_i \in [0, 1]$ with 1 indicating relevant and 0 indicating irrelevant.

Since we have a training set of queries used for obtaining the two classifiers, we can also train a regression function with parameters as the scores $R_{d_i}$ and $I_{d_i}$, to then obtain the notion of fuzzy relevance. Once we have this notion we try to use this fuzzy measure to weigh the documents differently, and thus treat them differently, which we describe in the next section.

## 11.3 Document Weighting

If we have an idea of the chance of a document being relevant i.e. its fuzzy relevance, then we can design a weighting scheme such that the documents whose relevance is more get a higher score. For example a simple weighting scheme is one where weights are given in proportion to the document's fuzzy relevance. Hence the more relevant it seems, the more importance it is given. Once we have these weights they can be easily integrated into most PRF methods, including the PIR2 method we discussed in chapter 9.

For example consider the MLE method of PRF, where we create a feedback model, which is simply the MLE estimate of the documents in the feedback set. As an alternative to this consider a weighted MLE model where instead of creating a simple MLE model, there is a weighted MLE, i.e. if a document $d_i$ has weight $w_i$ and if a term $t$ occurs in it $c$ times, then in the weighted MLE scheme this instead treated as the word occurring $w_i.c$ times. Since currently the irrelevance classifier and relevance classifiers have not been built, we are yet to test any weight functions.

Thus to gage the potency of the idea, consider Table 12.1. The values indicate how the performance of the MLE model increases on using a weighted scheme where the relevant documents are weighed twice as much as irrelevant documents.

| **ID** | **MLE** | | **Weighted MLE with 2:1 weights for Rel. Docs** | | |
|---|---|---|---|---|---|
| | MAP | P@5 | MAP | P@5 | % MAP Change |
| LAT | 0.438 | 0.49 | 0.474 | 0.538 | 8.2% |
| GH | 0.401 | 0.434 | 0.426 | 0.492 | 6.1% |
| AP | 0.317 | 0.483 | 0.332 | 0.557 | 4.7% |
| WSJ | 0.293 | 0.507 | 0.305 | 0.584 | 4.1% |
| SJM | 0.223 | 0.36 | 0.241 | 0.421 | 8.4% |

Table 11.1: Change in Feedback performance with $m$

Thus we see even with a simple 2:1 weighting scheme on a simple method like MLE there is a 4-9% improvement in MAP values across collections. Hence if the fuzzy relevance function is accurate enough then the document weighing scheme alone can lead to large improvements.

# Chapter 12

# Unified Framework

Thus we have seen different aspects of using irrelevance data to improve PRF in Chapter 8, 9, 10 and 11. In this chapter we present a framework to merge all these ideas together to obtain a highly-robust, stable and high-performing PRF system. In figure 12 we present this framework as a flowchart, signifying where the data flow occurs and the order in which the functions run.

The key components of the framework are:

- Initial Retrieval: This is the standard LM Ranking
- Pseudo-Irrelevancy Identification: This is the piece which will identify the pseudo-irrelevant set, as described in Chapter 8.
- High-Precision Irrelevancy/Relevancy Classifiers(**TIR System**): This is the piece which will learn the irrelevance and relevance classifiersas described in Chapter 10, and evict the irrelevant documents from the feedback set.
- Positive Resampling: This step follows the eviction of irrelevant documents from the feedback set. If there were too many irrelevant documents in the feedback set, then the residual feedback set may be too small. In this case we may instead sample documents from just

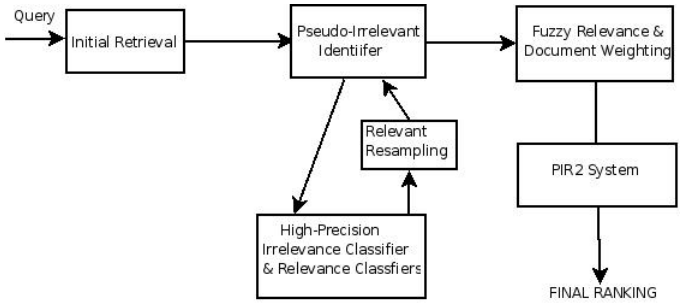


Figure 12.1: Unified Framework

outside the top $k$, and run the high-precision relevance classifier on them. If they are found to be relevant they are added to the feedback set.

- Fuzzy Relevance and Document Weighting: Once we have the feedback set, from which we are no longer able to evict any irrelevant documents from, we then will find the fuzzy relevance values for the documents in the feedback set, and accordingly find the document weights
- PIR2 System: The system as described in Chapter 9 which produces the final feedback model, and thus the ranked list.

A further explanation for some of the flows are:

- The TIR system links to the positive resampling system, as after eviction of irrelevant documents from the feedback set, the feedback set size may become too small. Hence we try to obtain any high-scoring relevant documents, which are just outside the top $k$.
- The TIR system links back to the pseudo-irrelevant identification system because once the feedback set, contains more relevant documents the identification algorithm starts performaning better, and we get higher recall and precision values. In table 12.1 we have provided an example of this, where we see the identification system's recall and the MAP values of the PIR2 system increase drastically, on removing the irrelevant documents from the feedback set.

  Hence the idea is that if this cycle runs 2-3 times, the composition of the pseudo-irrelevant set will be very high, along with the feedback set, being almost entirely composed of relevant documents. This apart from the many irrelevant documents we would have obtaned from the TIR system, and thus can eliminate.
- The identification system links to the Document Weighting Scheme, as only once the entire process of fixing the feedback set is finished, can the documents be weighed.
- PIR2 System is the final system, as it is the ranking system. It can directly incorporate the weights given by the weighting system as well, and thus produce the final ranked list.

| **ID** | **Original Values** | | **Values after identifying Irrel Docs** | |
|---|---|---|---|---|
| | Recall | MAP | Recall | MAP |
| LAT | 82.8% | 0.452 | 96.7% | 0.58 |
| GH | 82.7% | 0.419 | 96.2% | 0.511 |
| AP | 81.8% | 0.339 | 96.3% | 0.386 |
| WSJ | 82% | 0.311 | 95.5% | 0.347 |
| SJM | 80% | 0.243 | 94.8% | 0.295 |

Table 12.1: Change in Values after Identifying Irrelevant Documents in top $k$

# Chapter 13

# Discriminative Methodology of IR

In this chapter, we describe some of our work which was done while participating in the Adhoc Information Retrieval and *Cross Lingual Information Retrieval* (CLIR) tasks of FIRE (*Forum for Information Retrieval Evaluation*) 2010. While all the work in this thesis, discusses prior to this, are in the generative framework of information retrieval, in this chapter we explore a discriminative approach to IR.

In our approach, besides the basic term based matching features proposed in literature, we include novel features which model certain crucial elements of relevance like named entities, document length and semantic distance between query and document. We also investigate the effect of including Pseudo-Relevance Feedback (PRF) based features in the above framework.

For the CLIR tasks, while using the same features as those mentioned above, we use a query translation approach which uses bi-lingual dictionaries. For query disambiguation, we use an iterative query disambiguation algorithm. In the disambiguation approach, in lieu of regular term-term co-occurrence measures proposed in literature, we use the similarity between terms in LSI space.

## 13.1 Previous Work

Various approaches have been used for IR. These include, Vector Space Models, Generative approaches and Discriminative approaches. Any combination of arbitrary features can be included in the discriminative framework. This allows exploring use of different features representing the importance of document. Relative weights of these features can be learnt using discriminative models like Support Vector Machines (SVMs). Thus the discriminative framework, tends to perform well (as seen in the case of Web Search where most existing systems are discriminative in nature), but its' working is difficult to understand, and the learned feature weights difficult to intuitively understand. Nallapati [Nal04] explored the applicability of discriminative classifiers for IR. He tried to solve the problem of retrieval by changing it to a classification problem. He also proposed a set of 6 robust and well-thought-out features, which indirectly capture the scores from different scoring function such as KL-Divergence based ranking, tf-idf scores and others such. These features are

described in the next section.

Joachims et. al.[Joa02] introduced RankSVMs, which are a modification of the SVM formulation, tailored for the task of Information Retrieval. This tries to learn feature weights, while trying to rank the documents as per a list itself. In our experiments, we explore the use of RankSVMs (discriminative models) for monolingual as well as cross lingual retrieval. We use the statistical (tf, idf) features, as described in [Nal04] as the basic set of features. Along with this we have explored the use of features based on Co-ord factor, Document Length Normalization, NE features, Phrasal NE (*Named Entities*) and LSI (*Latent Semantic Indexing*) features.

## 13.2 Features Used

As mentioned we use the statistical features, proposed by Nallapati et. al, and experiment with some new features as well, related to the Co-ord Score, Document Length Normalization Factor, Named Entities Features and Term Similarity features

### 13.2.1 Statistical Features

We use 6 statistical features, which are accurately given in Table 13.1. These features are:

- *Term Frequency*:
  This feature measures the total number of term matches between a given query, $q$ and a given document, $D$.
- *Normalized Term Frequency*:
  This feature measures total number of matches between the query and the document and normalizes it with respect to the document length.
- *Inverse Document Frequency*:
  This measures the importance of a document in terms of Inverse Document Frequency ($IDF$), which is a measure of importance, for each query term, which overlaps with the document content.
- *Normalized Cumulative Frequency*:
  This measures the overall frequency of the overlapping terms (between a query and a document) in the whole corpus
- *IDF-Weighted Normalized Term Frequency*:
  This is similar to the normalized term frequency, except that each term is weighted by its importance, measured as IDF.
- *Cumulative Frequency-Weighted Normalized Term Frequency*:
  This is similar to the above feature, replacing the weighting factor by the cumulative frequency, as described in Normalized Cumulative Frequency Feature.

| Sr. No. | Feature Name | Description |
|---|---|---|
| 1 | Term Frequency | $\sum_{q_i \in Q \cap D} log(c(q_i, D))$ |
| 2 | Normalized Term Frequency | $\sum_{q_i \in Q \cap D} log(1 + \frac{c(q_i,D)}{\|D\|})$ |
| 3 | Inverse Document Frequency | $\sum_{q_i \in Q \cap D} log(idf(q_i))$ |
| 4 | Normalized Cumulative Frequency | $\sum_{q_i \in Q \cap D} log(\frac{\|C\|}{c(q_i,C)})$ |
| 5 | Normalized Term Frequency, weighted by IDF | $\sum_{q_i \in Q \cap D} log(1 + \frac{c(q_i,D)}{\|D\|} idf(q_i))$ |
| 6 | Normalized Term Frequency, weighted by cumulative frequency | $\sum_{q_i \in Q \cap D} log(1 + \frac{c(q_i,D)}{\|D\|} \frac{\|C\|}{c(q_i,C)})$ |

Table 13.1: Statistical Features

### 13.2.2 Co-ord Factor

For a given query $q$ and document $D$, *Co-ord Score* is a scoring factor which depends on how many of the query terms are found in the specified document. Typically, a document that contains more of the query's terms will receive a higher score than another document with fewer query terms. This is one of the measures, used internally by the ranking system of the popular IR tool: *Lucene*, which thus inspired us to incorporate it in our approach, as it enable us to study the importance of the factor as compared to other factors, and thus indirectly the potency of this measure.

### 13.2.3 Document Length Normalization

It is well known that retrieval algorithms are usually biased towards shorter documents [SBMM96]. Well-established algorithms such as BM-25, perform well because they rectify this problem. Hence we felt it was an important factor to consider, and thus incorporate a factor to remove this bias. We thus have a feature (corresponding to the pivoted normalization, as described in [SBMM96]) as the smoothing factor.

### 13.2.4 Named Entity Features

Another set of novel features introduced by us, are the Named Entity features. We observed that most of the important words in the queries were Named Entities. We felt that, in a given document, the presence of query words which are named entities, is a strong sign of an important document. We capture this intuition by incorporating this as a feature. This feature calculates sum of term frequency of Named Entity query words in document. Sometimes complete Multi-Word Named Entity are not present in the same form as in the query, *i.e.*, they may not occur in the document continuously. To allow for this we resort to using phrasal NEs. We check to see if all the Named Entity words, which are present in the query, are also present within a fixed context window of the document.

Consider as example the query "Laloo Prasad Yadav". For this query, it is not necessary that all words of the NE (*i.e.* Laloo Prasad Yadav) may not be found consecutively in the document.

However, the presence of all 3 words in a context window of size 50, is a strong indicator of the document pertaining to the given NE, and thus being relevant to the query. Hence using a Phrasal scoring system, is a better option in this example. To compute Phrasal Scores, we use the score provided by Lucene's Phrase Query Scoring.

### 13.2.5 LSI Feature

Latent Semantic Indexing (LSI) is an indexing method that uses a mathematical technique called Singular Value Decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.

This feature captures the LSI (*Latent Semantic Indexing*) based similarity between the given query and a document. This was implemented using the "Semantic Vectors Package" [WF08]. Instead of word-based match between query and document, as captured by the "Term Frequency" feature, this feature measures the concept-based match between query and document. In other words, instead of looking for a overlap in the Term-space, this looks for overlap in the Concept-space, where the terms are mapped to concepts, as defined in the LSI algorithm.

### 13.2.6 PRF Features

Unlike the Language Modeling Framework, there is no clean way of integrating PRF into the discriminative framework. In our approach, we considered the Top 20 terms, as predicted by the Zhai-Lafferty MBF approach, which has been discussed earlier in Section 3.2. We then append these terms to the query, *i.e.* we compute the values for the statistical features proposed earlier, and use them as features as well.

## 13.3 Cross Lingual IR

Our CLIR system is based on a dictionary based approach to query translation. After removing the stops words from the query, the query words are passed to the stemmer. We used the Hindi and Marathi stemmers developed at CFILT, IIT Bombay. The stemmed query words are translated using the Hindi and Marathi dictionary. We used the Hindi dictionary available at "*www.cfilt.iitb.ac.in*" for performing Hindi-English translation. This dictionary contains 131750 entries. The corresponding Marathi-English bilingual dictionary contains 31845 entries.

<topics>
<top lang="hi">
<num> 76 </num>
<title> *gurjaron aur meena samuday ke beech sangharsh*</title>

Table 13.2: FIRE 2010 Topic Number 76

| **Language** | **No. of Documents** | **No. of Tokens** | **No. of unique terms** | **Avg. Doc. Length** |
|---|---|---|---|---|
| English | 125586 | 33259913 | 191769 | 264.84 |
| Hindi | 95216 | 38541048 | 146984 | 404.77 |
| Marathi | 99275 | 25568312 | 609155 | 257.55 |

Table 13.3: Details of FIRE 2010 Corpus

| **S.No.** | **Description** | **Run ID** |
|---|---|---|
| 1 | EN Monolingual Title without Feedback | IITBCFILT_EN_MONO_TITLE_BASIC |
| 2 | EN Monolingual Title with Feedback | IITBCFILT_EN_MONO_TITLE_FEEDBACK |
| 3 | EN Monolingual Title+Desc without Feedback | IITBCFILT_EN_MONO_TITLE+DESC_BASIC |
| 4 | EN Monolingual Title+Desc with Feedback | IITBCFILT_EN_MONO_TITLE+DESC_FEEDBACK |
| 5 | HI Monolingual Title without Feedback | IITBCFILT_HI_MONO_TITLE_BASIC |
| 6 | HI Monolingual Title+Desc without Feedback | IITBCFILT_HI_MONO_TITLE+DESC_BASIC |
| 7 | HI Monolingual Title+Desc with Feedback | IITBCFILT_HI_MONO_TITLE+DESC_FEEDBACK |
| 8 | MR Monolingual Title without Feedback | IITBCFILT_MR_MONO_TITLE_BASIC |
| 9 | MR Monolingual Title with Feedback | IITBCFILT_MR_MONO_TITLE_FEEDBACK |
| 10 | MR Monolingual Title+Desc without Feedback | IITBCFILT_MR_MONO_TITLE+DESC_BASIC |
| 11 | MR Monolingual Title+Desc with Feedback | IITBCFILT_MR_MONO_TITLE+DESC_FEEDBACI |
| 12 | HI-EN Bilingual Title without Feedback | IITBCFILT_HI_EN_TITLE_BASIC |
| 13 | HI-EN Bilingual Title with Feedback | IITBCFILT_HI_EN_TITLE_FEEDBACK |
| 14 | MR-EN Bilingual Title without Feedback | IITBCFILT_MR_EN_TITLE_BASIC |
| 15 | MR-EN Bilingual Title with Feedback | IITBCFILT_MR_EN_TITLE_FEEDBACK |

Table 13.4: Details of Monolingual and Bilingual Runs Submitted

### 13.3.1 Query Transliteration

Many proper nouns of English like names of people, places and organizations, used as part of the source language query, are not likely to be present in the bi-lingual dictionaries. Table 13.2 presents a sample Hindi topic from FIRE 2010. In the above topic, the word is "Gurjars" written in Devanagari. Such words are to be transliterated to generate their equivalent spellings in the target language. Since Hindi and Marathi use Devanagari which is a phonetic script, we use a Segment Based Transliteration approach for Devanagari to English transliteration. The current accuracy of the system is 71.8% at a rank of 5[CD09].

### 13.3.2 Translation Disambiguation

Given the various translation and transliteration choices for each word in the query, the aim of the Translation Disambiguation module is to choose the most probable translation of the input query Q. The main principle for disambiguation is that the correct translation of query terms should co-occur in target language documents and incorrect translations will not co-occur. Given the possible target equivalents of two source terms, we can infer the most likely translations by looking at the pattern of co-occurrence for each possible pairs of definitions. For example, for a query "*nadi jal*", the translation for "*nadi*" is {river} and the translations for "*jal*" are {water, to burn}. Here, based on the context, we can see that "water" is a better translation for the second word "*jal*", since it is more likely to co-occur with river.

Assuming we have a query with three terms $s_1, s_2$ and $s_3$ each with different possible translations/transliterations, then the most probable translation of the query is the combination amongst the possibilities which has the maximum number of occurrences in the corpus. We use a page-rank style iterative disambiguation algorithm proposed by Christof Monz et. al. [MD05] which examines pairs of terms to gather partial evidence for the likelihood of a translation in a given context. We finally use a Latent Semantic Indexing (LSI) [DDF$^+$90] based similarity measure between two terms. We use this feature as a link weight in Iterative disambiguation algorithm.

## 13.4 Experiments and Results

This approach was used in the FIRE 2010 task. The details of the FIRE 2010 document collection are given in Table 13.3. We used TREC Terrier [OAP$^+$06] as the monolingual English IR engine. We initially experimented with SVM and Rank-SVM, and found that Rank-SVM always performs better than SVM. Hence, we use RankSVM in all our experiments with the features mentioned above. The documents were indexed after stemming and stop-word removal. As a training set, we used the FIRE-2008 topic set, which consists of 50 topics each, in Hindi, Marathi and English along with their relevance judgments.

We used the Hindi and Marathi stemmers and morphological analyzers developed at CFILT, IIT Bombay for stemming the topic words. For English, we used the standard Porter stemmer.We submitted the Title(T), and Title + Description (T+D) runs for all the tasks. The details of the runs which we submitted are given in Table 13.4. Results of monolingual runs are shown in the Table 13.5 and cross-lingual runs are given in the Table 13.6. We used the following standard evaluation measures [MRS08]: Mean Average Precision (MAP), Precision at 5, 10 and 20 documents (P@5, P@10 and P@20) and Recall.

## 13.5 Discussion

In this approach, we introduced four new features, NE feature, Co-ord factor, Normalization and LSI based similarity to see the impact they have on retrieval. We describe the results obtained

| Run ID | MAP | P@5 | P@10 | P@20 | Recall |
|---|---|---|---|---|---|
| IITBCFILT_EN_MONO_TITLE_BASIC | 0.3803 | 0.4280 | 0.3540 | 0.2850 | 0.9832 |
| IITBCFILT_HI_MONO_TITLE_BASIC | 0.2384 | 0.3360 | 0.2760 | 0.2200 | 0.8175 |
| IITBCFILT_HI_MONO_TITLE+DESC_BASIC | 0.3317 | 0.4440 | 0.3680 | 0.2940 | 0.8962 |
| IITBCFILT_MR_MONO_TITLE_BASIC | 0.2135 | 0.2560 | 0.2380 | 0.1850 | 0.8019 |
| IITBCFILT_MR_MONO_TITLE+DESC_BASIC | 0.2399 | 0.2800 | 0.2720 | 0.2280 | 0.9275 |

Table 13.5: FIRE 2010 Monolingual Results

| Run ID | MAP | P@5 | P@10 | P@20 | Recall |
|---|---|---|---|---|---|
| IITBCFILT_HI_EN_TITLE_BASIC | 0.2848 | 0.2960 | 0.2580 | 0.2130 | 0.9081 |
| IITBCFILT_HI_EN_TITLE_FEEDBACK | 0.3365 | 0.3560 | 0.3160 | 0.2510 | 0.9142 |
| IITBCFILT_MR_EN_TITLE_BASIC | 0.2515 | 0.2776 | 0.2286 | 0.1878 | 0.7850 |
| IITBCFILT_MR_EN_TITLE_FEEDBACK | 0.2771 | 0.2939 | 0.2510 | 0.2031 | 0.8411 |

Table 13.6: FIRE 2010 Cross Lingual Results

| Language | Basic | Basic + NE | Basic + LSI | Basic + Norm | Basic + Co-ord |
|---|---|---|---|---|---|
| English | 0.3715 | 0.3759 | 0.3818 | 0.3689 | 0.3707 |
| Hindi | 0.2247 | 0.1900 | NA | 0.2284 | 0.2338 |
| Marathi | 0.2141 | 0.2142 | NA | 0.2128 | 0.2148 |

Table 13.7: FIRE 2010 Cross Lingual Results

after adding these features to the Basic Feature Set (Statistical Features) for different languages. The table 13.7 shows the MAP score with different features for different languages. These scores represent the MAP values obtained by using only the "Title" of the query.

In English, we observe that the LSI feature helps in improving the performance. However, the NE and Document Length Normalization features are not useful. In Hindi, we observe that Co-ord factor and document length normalization improve the results. Finally, in Marathi the extra features do not improve performance over basic term features. Initial analysis into the failure of these features reveal that many of these problems are due to inaccuracies of the stemmer. However, a thorough error-analysis needs to be done to confirm the above hypothesis.

## 13.6 Some Preliminary Conclusions

We presented the evaluation of our Hindi-English, Marathi-English CLIR systems performed as part of our participation in FIRE 2010 Ad-Hoc Bilingual task. In CLIR, we used a LSI based term-term similarity metric for query disambiguation. We also discussed the monolingual evaluation of our system for English, Hindi and Marathi languages using a discriminative approach to IR. Besides the regular term based features, we experiment with novel features like NEs, LSI based similarity features, document length normalization, co-ord factor, pseudo-relevance feedback and

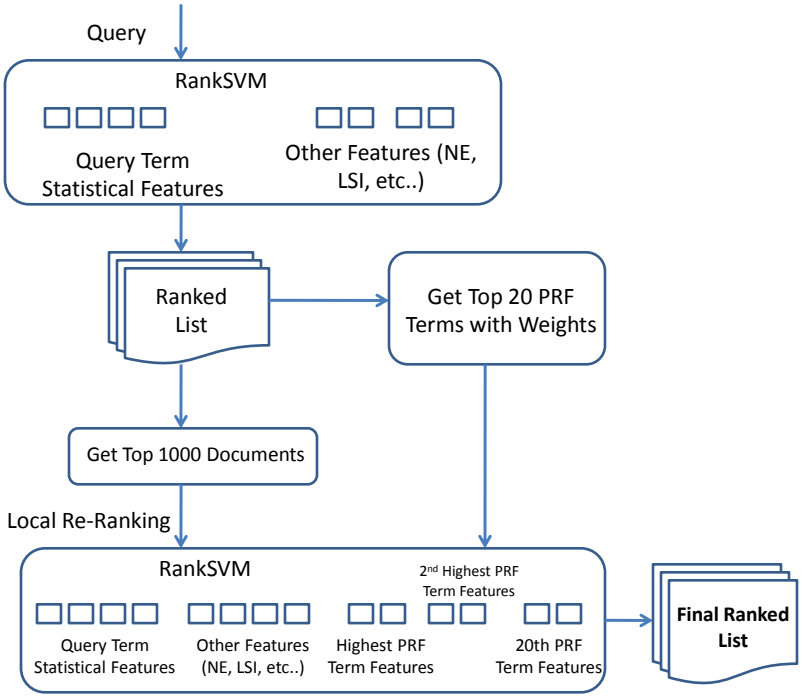


Figure 13.1: Schematic of the Proposed Discriminative Approach

studied their impact. Results show improvement in some languages but they are not consistent and a detailed error analysis is required to understand the behavior of each of these features.

## 13.7 Ways of Incorporating PRF in the Discriminative Framework

Although the approach we proposed, provides for a principled method of integrating PRF into the discriminative framework, it still has its' flaws. It is very simplistic and preliminary results, as seen in the CLIR setting, still show the performance to be some way off that of the Language Model-based systems. We propose a new approach, which depends on the concept of local re-ranking. *Local Re-ranking* refers to the concept of using the ranked list of a first pass retrieval, and using a new ranking function to re-order the top 1000 documents of the first-pass retrieval. This way, the recall at 1000 stays the same, but you can hope to significantly improve precision levels.

The inspiration for this method, came from the observation that the current method of feedback, did not lead to large changes in recall, as seen in Table 13.6. Hence instead of focusing on improving both precision and recall, we target making significant improvements in precision alone. Thus the idea, here is to use a RankSVM like approach twice. In the first stage, we run RankSVM with no PRF features included. Once we have the ranked list for this first stage, we collect the Top 1000 documents, and perform another RankSVM, which now involves the PRF features, on this reduced set of 1000 documents.

The problem with performing the second step on the whole collection, is the choice of -ve

examples, when it comes to training. Since the -ve examples provided by the relevance judgements, are when only the query is used, and not the expanded query, using the given -ve examples as -ve examples for training the RankSVM running on the expanded query, results in very poor performance. Hence we have to resort to local re-ranking, as we have a simpler way of training the second RankSVM. The training set is first split into two sets: The first set is used to train the first RankSVM, which does not use any PRF features. The model thus learnt is then run on the queries in the second half of the training set. For each of these queries, the top 1000 documents are taken, and the second RankSVM (which includes separate PRF features for each of the 20 expansion terms) is trained using the known +ve and -ve examples from the top 1000 as data points.

For testing, as mentioned before, two RankSVM methods need to be run, to get the final ranked list of documents. The detailed System Architechture is given in Figure 13.1. This method of performing feedback is much more principled, while maintaining the simplicity in the manner of finding PRF terms. Here since the PRF terms do not share a common feature, there is some distinction made between the most important PRF term and the least important which was missing in the previous formulation. Since this method improves upon the design flaws of the previous method, without introducing any new ones,it is expected to perform better.

# Chapter 14

# Putting Things Together

As part of this thesis, we have seen many different methods of performing PRF. In this chapter we review these different ideas and try to contrast them with one another. We also try to see if we can combine some of them, to get a better system.

## 14.1 Summary of Methods

We have broadly seen 4 methods of performing Query Expansion, 3 of which do so by trying to improve PRF. These methods can be summarized as:

1. *Language-Assisting Methods*: We out forth a set of approaches, within the Language-Modeling Framework, wherein we used the assistance of one language, to improve PRF in another. We also saw extensions to this method, which incorporated multiple assisting languages, and combined evidence from all of them to help improve PRF. The positives of these methods can be summarized as follows:

   - Mitigated the 2 well-known problems of PRF. Introduced synonyms and morphological variants by using a probabilistic dictionary. Also made PRF more robust, by doing a Risk-Minimization across collections.
   - Clean Framework proposed, to choose models according to the queries.
   - Allowed for PRF of weaker languages to be improved. As seen in the analysis of the results, the improvements were largest, when the source language had poor performance.
   - Potentially very useful in the Web Scale due to the difference in coverage across languages.

   At the same time, the method also has some drawbacks. Namely:

   - Resource-Heavy. Unlike the other methods discussed in this thesis, this approach requires comparable collections across languages. It also needs a good query translation engine. Lastly it requires parallel corpus to learn the dictionaries.

- Not all languages, can assist each other. Thus only certain languages can be used to assist.

On the whole, this method was shown to perform consistently and gave significant improvements over the baselines, with further extensions also proposed.

2. *Irrelevance-Based Methods*: We proposed a set of methods, again within the Language-Modeling Framework, which directly attacked the most serious problem plaguing PRF, *i.e.* irrelevant documents in the feedback set. Along the way we defined the concept of pseudo-irrelevance, and described a method to extract these documents. We then defined frameworks which used this information to improve PRF. We also used these documents, to build a classifier which was able to identify irrelevant documents in the feedback set. We also proposed extensions which unified all the systems, into a single framework. The positives of this method can be described as:

   - Proposed a neat, clean framework which integrated all the ideas.
   - Directly solved the problem of PRF, thus leading to better robustness
   - Defined the concept of pseudo-irrelevance.

   One drawback of this approach is that there are lot of corpus-specific parameters, which will have to be learnt when testing on a new corpus. Another drawback is that since identifying irrelevant documents was shown to be hard, there is only a limited point till where this method can be taken.

3. *Random-Walk Based Methods*: We propose a method of performing Query Expansion in the LM framework, where we used a statistical thesaurus to find synonyms of query terms and other semantically related terms. We also proposed a method of learning this thesaurus by performing a random walk across terms in a language. We also proposed extensions to the Random Walk procedure to reduce noise. The biggest positive of this method is the introduction of Lexically and Semantically related terms using the thesaurus, along with the method proposed to learn the thesaurus. However this method suffers the big drawback of being a variation of synonym-based expansion methods, and thus not learning co-occurrence based expansion terms, which automatically limit performance.

4. *Discriminative Methods*: As part our participation in FIRE 2010, we also studied methods in the discriminative framework. We proposed extensions to current approaches, which incorporated Named Entity Features, Term/Concept Similarity Features and tried to integrate PRF into this approach as well. We also proposed extensions, which improved the method of integrating PRF in this framework. The biggest positives in this method were:

   - It allowed for a lot of extra information, such as Named Entities and Term/Concept Similarities to be neatly introduced in the ranking function, which is not possible in any of the previous settings.
   - Can leverage optimizations from the Web Scale approaches, since these are the family of methods used on the Web Scale. Can also allow to add more features, without much change in the system. Thus easily extendable.

   However a big drawback here was that the performance did not match those of the other methods. Also the method of obtaining the PRF-based expansion terms, was not designed for this setting. Hence a more natural way of obtaining expansion terms is required.

## 14.2 Can They be Combined?

Though all the methods were proposed independently, the question arises if some of them can be clubbed together, to create a much better system. We look into two possible combinations:

- The MultiPRF methods and its' variants all use the PRF module as a plug-and-play. In other words, there is no limitation to the PRF method used, and thus methods apart from the simple Zhai-Lafferty provided MBF can be used. Thus the irrelevance-based methods can be neatly combined with this approach, as they are another form of PRF methods in themselves.
- Similarly the Discriminative methods, simply use the expansion terms of a PRF method, without being influenced by what the method is. Hence the MultiPRF or Irrelevance-based methods can be used as the native PRF methods and thus combined with these approaches.

Thus in this manner we have seen how some of these methods can be integrated to perform better.

# Chapter 15

# Conclusions

The main purpose of this work is to use irrelevance data to improve the stability and performance of PRF. We have shown how PRF is affected severely due to the presence of irrelevant documents in the feedback set, and hence these irrelevant documents needs to be removed. Some of the key points that we have presented in this thesis are:

- We presented a novel approach to PRF called Multilingual PRF in which the performance of PRF in a language is improved by taking the help of another language collection. We also showed that MultiPRF addresses the fundamental limitations of monolingual PRF, *viz.*, (i) the inability to include term associations based on lexical and semantic relationships and (ii) sensitivity to the performance of the initial retrieval algorithm. Experiments on standard CLEF collections across a wide range of language pairs with varied degree of familial relationships show that MultiPRF consistently and significantly outperforms monolingual PRF both in terms of robustness and retrieval accuracy. Our error analysis pointed to obvious contributing factors to error: (i) inaccuracies in query translation including the presence of out-of-vocabulary terms, (ii) poor retrieval on English query, and in a few rare cases, (iii) inaccuracy in the back translation.
- We analyzed the sensitivity of the MultiPRF algorithm to the different components it relies on, such as the Query Translation accuracy and the Assisting Language Used. We also showed that the gains in MultiPRF cannot be replicated by using another collection in the same language, or by using a thesaurus based expansion method directly or in combination.
- Extended the MultiPRF approach, and allowed for multiple assisting languages in the MultiAssistPRF approach. We argued why using more than 1 assisting language, was likely to make PRF more robust, while still obtaining Lexically and Semantically Related Terms. We also studied a framework of combining the different feedback models by using a classifier.
- We studied a framework of query expansion, which used a probabilistic thesaurus. We also proposed a method of learning such a thesaurus by performing a random walk across terms in one language to terms in another. We also studied possible improvements of this method as well.

- Defined the concept of pseudo-irrelevance, analogous to the concept of pseudo-relevance. We presented a method to identify the pseudo-irrelevant documents, which were high-scoring documents that are unlikely to be relevant. We then presented the performance of this identification method.

- Discussed how the pseudo-irrelevance data can be used. Proposed two different methods of using these documents. The first method was a discriminative method where we tried to identify discriminative terms, as terms which helped differentiate between the pseudo-relevant documents and the pseudo-irrelevant documents. We then looked at another method, which introduced the concept of Query-Specific noise. The method, which is an extension of the Model-Based Feedback algorithm, tried to use the pseudo-irrelevant documents to obtain an estimate of the query-specific noise, which is then filtered from the feedback documents to obtain the final feedback model. We analyzed the performance of the two methods, and saw the variations across collections.

- We then looked at how irrelevant documents in the feedback set, significantly reduce performance thus limiting the deployment of PRF. We saw the different reasons why identifying these irrelevant documents is difficult, and proposed a set of properties which can help in their identification. Finally we put forth the idea of learning a high-precision irrelevance classifier, using a set of features, as well as the algorithm to obtain the feature weights, and then identify these documents. We then also looked at some questions which arose from this avenue, and also looked into the idea of learning a high-precision relevance classifier.

- We then looked at the notion of fuzzy relevance, to assist us in the cases where a document cannot be accurately identified as either irrelevant or relevant. We saw a way of learning the fuzzy relevance value of a document using the information from the relevance and irrelevance classifiers. We then saw a method to utilize these fuzzy relevance values, in a document-weighting scheme which can be easily extended to most PRF schemes.

- We then saw a unified framework, which unified all the previous ideas, into a single, clean framework. We saw the possible benefits of linking two different components, and how some of the components when used with others, improved values.

- Next, we saw a different set of retrieval models: the discriminative models. We studied some previous work in this field and then proposed some new ideas which are unique to the discriminative framework. We tried to add features incorporating Named Entities, Concept Similarities and proposed a method of involving PRF in this framework.

- Finally, we saw a comparison between different methods, with pros and cons of each identified. We also studied possible ways of combining different ideas, to further gains.

# Appendix A

# Settings for Irrelevance-Based Experiments

In this appendix we discuss the settings for some of our experiments, along with the datasets used, and the framework of the retrieval process. The settings used for the Language-Model based experiments differ from those used for the Discriminative-Models Based Experiments, and hence we detail each of them separately.

## A.1 Language-Model Based Experiments

### A.1.1 Test Collections

We have used TREC [1] and CLEF [2] collections for all experiments in this part of our work. We refer to each of the collections by the ID given in the first column of Table A.1. Note that in all our experiments we exclude topics for which there are no relevant documents provided in the corresponding qrels file. Also note that in all our experiments we use only the Topic of the query.

### A.1.2 General Setup

The IR engine we used for all Language-Model based experiments was IRawata [Bho08]. We first remove stop-words, and then stem queries and documents using the Porter stemmer. We used KL-Divergence based retrieval with two-stage Dirichlet smoothing as the initial retrieval. We used Model-Based Feedback as the representative PRF technique, with parameter settings of $\alpha = 0.5$ and $\lambda = 0.5$. The feedback set we used for PRF was the top 10 documents from the initial retrieval. We use Mean Average Precision(MAP); Precision at 5(P@5) and Precision at 10(P@10) as our main performance indicators.

[1] http://trec.nist.org
[2] http://www.clef-campaign.org

| Coll. ID | Description | Source | No. of Topics |
|---|---|---|---|
| LAT | LA Times 1994 | CLEF '00 - '02 | 122 |
| GH | LA Times 1994 + Glasgow Herald 1995 | CLEF '03,'05 and '06 | 152 |
| AP | Associated Press '88,'89 | TREC Disk 1,2 | 149 |
| WSJ | Wall Street Journal | TREC Disk 1,2 | 150 |
| SJM | San Jose Mercury News | TREC Disk 3 | 94 |

Table A.1: Test Collections Used

### A.1.3 LibLinear

For the binary classifications tasks in our experiments we used LibLinear[FCH+08]. This is a library which allows for different kinds of classifiers to be learnt. We have used the default settings for LibLinear unless explicitly mentioned otherwise.

## A.2 Discriminative-Model Based Experiments

### A.2.1 Test Collections

We have used the FIRE [3] collections for all experiments in this part of our work. Collections are referred to by the ID mentioned in the first column of Table A.2. Note that this corpora is across different languages, and is a comparable corpora, i.e. Collections in different languages contain documents about similar topics. Here too we have used Topic of query only.

| Coll. ID | Collection Language | Description | Source | Topics |
|---|---|---|---|---|
| FENG | English | Telegraph | FIRE 2008 | 50 |
| FHIN | Hindi | Jagran | FIRE 2008 | 50 |
| FMAR | Marathi | Marathi News | FIRE 2008 | 50 |

Table A.2: FIRE Test Collections

### A.2.2 General Setup

For these experiments we have used the Terrier [4] IR engine. One of the reasons for using Terrier is its' support for many European languages(i.e. it provided stemmers and stop-word lists), and as our future work would involve testing on European languages, we chose to use it. We have used

[3] http://www.isical.ac.in/ clia/
[4] http://ir.dcs.gla.ac.uk/terrier/

SVMRank and SVMLite, as an implementation of RankSVM. We have used the default parameter settings for these, unless explicitly mentioned otherwise.

Here to compare the initial retireval performance of our method we have implemented KL-Divergence based retrieval with two-stage Dirichlet smoothing and also implemented BM25, one of the most standard retrieval algorithms, with default parameter settings. For comparing PRF performance we have used Model-Based Feedback as the representative PRF technique, with parameter settings of $\alpha = 0.5$ and $\lambda = 0.5$. The feedback set we used for PRF was the top 10 documents from the initial retrieval. We use Mean Average Precision(MAP); Precision at 5(P@5) and Precision at 10(P@10) as our main performance indicators.

# Appendix B

# Settings for Multilingual PRF-Based Experiments

This appendix briefly summarizes some of the experimental settings used, in all experiments related to the MultiPRF approach. We used the standard CLEF evaluation data in seven languages - Dutch, German, English, French, Spanish, Finnish and Hungarian using more than 600 topics. To begin. The details of the collections used in our experiments and their corresponding topics are given in Table 4.3. While choosing an assisting collection from a language which has multiple collections, we pick the collection such that the coverage of topics, in that collection, is similar to that of the original corpus, on which performance is being measured, so as to get meaningful feedback terms. As queries, we only use the *title* field of the topics. We also ignore the topics which have no relevant documents.

As a platform for our experiments, we used the Terrier platform to perform all our experiments, due to its' support of European Languages. We perform standard tokenization, stop word removal and stemming using the Porter Stemmer for English and the Snowball stemmers for the other languages. We report results only on standard evaluation measures namely *MAP*, *P@5*, *P@10* and *GMAP*.

The probabilistic bi-lingual dictionaries used in all our experiments, were learnt automatically by running GIZA++ on a parallel sentence aligned corpora. For all language pairs, except those involving Hungarian, we used the *Europarl Corpus* and in case of Hungarian-English, we used the Hunglish Corpus. Note that, since in IR we only deal with stemmed terms, the bilingual dictionaries also contain only stemmed terms. Hence we first tokenized and did all the required pre-processing, including stemming, on the Europarl corpora, before we ran GIZA++.

For the initial retrieval performance of our method we have implemented KL-Divergence based retrieval with two-stage Dirichlet smoothing. We use the MBF approach, as explained in previous chapters, as the feedback baseline. We tune the parameters of MBF, specifically $\lambda$ and $\alpha$, and choose the values which give the optimal performance on a given collection. We uniformly choose the top $k$ documents as the feedback set, with $k$ set as 10.

# References

[ACR04] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness, and selective application of query expansion. In *ECIR*, pages 127–137, 2004.

[Bho08] Abhijit Bhole. Pseudo-relevance feedback in information retrieval. Undergraduate Thesis, 2008.

[BP04] Martin Braschler and Carol Peters. Cross-language evaluation forum: Objectives, results, achievements. *Inf. Retr.*, 7(1-2):7–31, 2004.

[BS98] Martin Braschler and Peter Schäuble. Multilingual information retrieval based on document alignment techniques. In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 183–197, London, UK, 1998. Springer-Verlag.

[BSAS94] Chris Buckley, Gerald Salton, James Allan, and Amit Singhal. Automatic query expansion using smart : Trec 3. In *Proceedings of The Third Text REtrieval Conference (TREC-3)*, pages 69–80, 1994.

[CD09] Manoj Kumar Chinnakotla and Om P. Damani. Character sequence modeling for transliteration. In *ICON 2009*, IITB, India, 2009.

[CKSB08] Soumen Chakrabarti, Rajiv Khanna, Uma Sawant, and Chiru Bhattacharyya. Structured learning for non-smooth ranking losses. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 88–96, New York, NY, USA, 2008. ACM.

[CNGR08] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, New York, NY, USA, 2008. ACM.

[CRB10a] Manoj Chinnakotla, Karthik Raman, and Pushpak Bhattacharya. Multilingual PRF: "English Lends a Helping Hand". In *SIGIR '10: Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[CRB10b] Manoj Chinnakotla, Karthik Raman, and Pushpak Bhattacharya. Study of Using Languages To Help Improve PRF in Other Languages. In *ACL '10*, 2010.

[CTC05] Kevyn Collins-Thompson and Jamie Callan. Query expansion using random walk models. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 704–711, New York, NY, USA, 2005. ACM.

[CTZC04] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. A framework for selective query expansion. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 236–237, New York, NY, USA, 2004. ACM.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[DIS91] Ido Dagan, Alon Itai, and Ulrike Schwall. Two languages are more informative than one. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 130–137, Morristown, NJ, USA, 1991. Association for Computational Linguistics.

[DLLL97] T. Susan Dumais, A. Todd Letsche, L. Michael Littman, and K. Thomas Landauer. Automatic cross-language retrieval using latent semantic indexing. In *AAAI Technical Report SS-97-05*, pages 18–24, 1997.

[Eft96] E. N. Efthimiadis. *Query expansion*, volume 31, pages 121–187. Annual Review of Information Science and Technology, 1996.

[FCH+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, August 2008.

[GBZ08] Wei Gao, John Blitzer, and Ming Zhou. Using english information in non-english web search. In *iNEWS '08: Proceeding of the 2nd ACM workshop on Improving non english web searching*, pages 17–24, New York, NY, USA, 2008. ACM.

[HBCb+07] Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondej Bojar. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL, Demonstration Session*, pages 177–180, 2007.

[HTH99] David Hawking, Paul Thistlewaite, and Donna Harman. Scaling up the trec collection. *Inf. Retr.*, 1(1-2):115–137, 1999.

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[JJJW99] P. Jourlin, S. E. Johnson, K. Spärck Jones, and P. C. Woodland. Improving retrieval on imperfect speech transcriptions (poster abstract). In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 283–284, New York, NY, USA, 1999. ACM.

[Joa02] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[Joa05] Thorsten Joachims. A support vector method for multivariate performance measures. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 377–384, New York, NY, USA, 2005. ACM.

[JR07] Thorsten Joachims and Filip Radlinski. Search engines that learn from implicit feedback. *Computer*, 40(8):34–40, 2007.

[JWR00] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808, 2000.

[KL] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, (1).

[KSKB09] Mitesh M. Khapra, Sapan Shah, Piyush Kedia, and Pushpak Bhattacharyya. Projecting parameters for multilingual word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 459–467, Singapore, August 2009. Association for Computational Linguistics.

[LC01a] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA, 2001. ACM.

[LC01b] Victor Lavrenko and W. Bruce Croft. Relevance Based Language Models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA, 2001. ACM Press.

[LCC02] Victor Lavrenko, Martin Choquette, and W. Bruce Croft. Cross-lingual relevance models. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 175–182, New York, NY, USA, 2002. ACM.

[LZ01] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, New York, NY, USA, 2001. ACM.

[LZ03] John Lafferty and Chengxiang Zhai. Probabilistic Relevance Models Based on Document and Query Generation. In *Language Modeling for Information Retrieval*, volume 13, pages 1–10. Kluwer International Series on IR, 2003.

[LZ09] Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, New York, NY, USA, 2009. ACM.

[MC07] Donald Metzler and W. Bruce Croft. Latent concept expansion using markov random fields. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318, New York, NY, USA, 2007. ACM.

[MD05] Christof Monz and Bonnie J. Dorr. Iterative translation disambiguation for cross-language information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 520–527, New York, NY, USA, 2005. ACM.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[MSB98] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–214, New York, NY, USA, 1998. ACM.

[MTdK09] Edgar Meij, Dolf Trieschnigg, Maarten Rijke de, and Wessel Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing & Management*, 2009.

[Nal04] Ramesh Nallapati. Discriminative models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, New York, NY, USA, 2004. ACM.

[OAP+06] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[ON03] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[PC98] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM.

[Phi05] Koehn Philipp. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.

[Rob06] Stephen Robertson. On gmap: and other transformations. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 78–83, New York, NY, USA, 2006. ACM.

[Roc71] J. Rocchio. *Relevance Feedback in Information Retrieval*, pages 313–323. The SMART Retrieval System, 1971.

[RSJ88] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. pages 143–160, 1988.

[RUBB10] Karthik Raman, Raghavendra Udupa, Pushpak Bhatacharyya, and Abhijit Bhole. On improving pseudo-relevance feedback using pseduo-irrelevant documents. In *ECIR '10*, 2010.

[Sal71] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[SBMM96] Amit Singhal, Chris Buckley, Mandar Mitra, and Ar Mitra. Pivoted document length normalization. pages 21–29. ACM Press, 1996.

[SC99] Fei Song and W. Bruce Croft. A general language model for information retrieval (poster abstract). In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 279–280, New York, NY, USA, 1999. ACM.

[SMK05] Tetsuya Sakai, Toshihiko Manabe, and Makoto Koyama. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):111–135, 2005.

[Tie01] Jrg Tiedemann. The Use of Parallel Corpora in Monolingual Lexicography - How word alignment can identify morphological and semantic relations. In *Proceedings of the 6th Conference on Computational Lexicography and Corpus Research (COMPLEX)*, pages 143–151, Birmingham, UK, 28 June - 1 July 2001.

[TLJ+07] Tuomas Talvensaari, Jorma Laurikkala, Kalervo Järvelin, Martti Juhola, and Heikki Keskustalo. Creating and exploiting a comparable corpus in cross-language information retrieval. *ACM Trans. Inf. Syst.*, 25(1):4, 2007.

[TZ06] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA, 2006. ACM Press.

[TZ07] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 295–302, New York, NY, USA, 2007. ACM.

[UBB09] Raghavendra Udupa, Abhijit Bhole, and Pushpak Bhattacharyya. "a term is known by the company it keeps": On selecting a good expansion set in pseudo-relevance feedback. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 104–115, Berlin, Heidelberg, 2009. Springer-Verlag.

[Voo94] Ellen M. Voorhees. Query Expansion using Lexical-Semantic Relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[Voo06] Ellen Voorhees. Overview of the trec 2005 robust retrieval track. In *E. M. Voorhees and L. P. Buckland, editors, The Fourteenth Text REtrieval Conference, TREC 2005*, Gaithersburg, MD, 2006. NIST.

[WF08] Dominic Widdows and Kathleen Ferraro. Semantic vectors: a scalable open source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

[WFZ07] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. Improve retrieval accuracy for difficult queries using negative feedback. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 991–994, New York, NY, USA, 2007. ACM.

[WFZ08] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. A study of methods for negative relevance feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 219–226, New York, NY, USA, 2008. ACM.

[WHJG08] Dan Wu, Daqing He, Heng Ji, and Ralph Grishman. A study of using an out-of-box commercial mt system for query translation in clir. In *iNEWS '08: Proceeding of the 2nd ACM workshop on Improving non english web searching*, pages 71–76, New York, NY, USA, 2008. ACM.

[XC00] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

[XFW02] Jinxi Xu, Alexander Fraser, and Ralph Weischedel. Empirical studies in strategies for arabic retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 269–274, New York, NY, USA, 2002. ACM.

[XJW09] Yang Xu, Gareth J.F. Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, New York, NY, USA, 2009. ACM.

[YFRJ07] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, New York, NY, USA, 2007. ACM.

[YGJ+10] Yisong Yue, Yue Gao, Thorsten Joachims, Olivier Chapelle, and Anne Ya Zhang. Learning more powerful test statistics for click-based retrieval evaluation. In *SIGIR '10: Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2010.

[Zha] ChengXiang Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, (3).

[ZHS09] Peng Zhang, Yuexian Hou, and Dawei Song. Approximating true relevance distribution from a mixture model based on irrelevance data. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 107–114, New York, NY, USA, 2009. ACM.

[ZL01] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA, 2001. ACM.

[ZL04] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[ZY09] Jinglei Zhao and Yeogirl Yun. A proximity language model for information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, New York, NY, USA, 2009. ACM.