SYSTEMATIC OPTIMIZATION OF SEARCH ENGINES FOR DIFFICULT QUERIES

BY

MARYAM KARIMZADEHGAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Doctoral Committee:

Associate Professor ChengXiang Zhai, Chair & Director of Research
Professor Jiawei Han
Assistant Professor Miles Efron
Doctor Andrew Tomkins, Google Research

Abstract

With the advent of Web, text information is being generated across the globe at an unfathomable rate and covering countless topics. This dramatic growth in text information and the increasing number of ways people can utilize it has influenced our daily lives in fundamental and profound ways. The widespread and multi-purposed use of search engines is one example of how transformed our lives have become in relation to text information.

Although the vast majority of people's information needs can be served very successfully by current search engines, there is still a considerable number of queries that even the best search engines perform poorly on. In this dissertation, we propose a method of optimizing a search engine to better handle such difficult queries, which addresses issues and opportunities at three different stages of an interactive search. Specifically, we propose to improve search quality for difficult queries by: (1) Bridging the vocabulary gap by defining a semantic smoothing language model. A query can be difficult because it does not contain the optimal choice of related terms, or lacks sufficient discriminative terms. In the pre-retrieval stage, using semantic smoothing during query formulation can mitigate the effect of those omissions. (2) Incorporating user negative feedback, i.e., improving a search engine by learning from user feedback in the post-retrieval stage. When a query is so difficult that all the top retrieved results (e.g., top 10) are completely irrelevant, the feedback that a user can provide is solely negative. We propose a generalized optimization framework to learn from feedback on non-relevant documents to prune extensively, but carefully, a large number of non-relevant documents from the top of the ranking list. (3) Balancing priorities when learning from user interaction in order to optimize the whole session. When presenting the search results to the user, there is a tradeoff between promoting those with the highest immediate utility and promoting those with the best potential for collecting feedback information, which can be used to better serve the user over the course of the session (interactive search stage). We frame this tradeoff as a problem of optimizing the diversification of search results, and we propose a machine learning approach that adaptively optimizes each individual user query such that we maximize the overall utility of the entire session.

In summary, this dissertation is expected to advance the state of the art of search engines by providing

a suite of novel search algorithms, which use the listed approaches to improve a search engine's ability to handle difficult queries.

To my parents and my sisters

for their unconditional love and support

Acknowledgments

There are many people who have influenced my life as a PhD student. First and foremost, I am so grateful to have the opportunity to work with my advisor, Professor ChengXiang Zhai. Cheng taught me many things including the importance of aiming for high standards, the art of finding interesting research problems, and the way of developing elegant research solutions. I am inspired by his brilliance, persistence, hard work and dedication. I am deeply grateful to him for his constructive technical guidance and for his encouragement; this thesis would not have been possible without his constant guidance and encouragement. Working with him over the years of my doctoral study has been an honor and a joy for me.

I would also like to thank Professor Geneva Belford for all her help, support and guidance since the very first year of my PhD. I am eternally grateful to her for her advice regarding my career.

My gratitude extends to all the other thesis committee members. My special thanks to Professor Miles Efron for giving me the opportunity to discuss several different topics with him, for his insightful comments and for his support. I am also thankful to Professor Jiawei Han for the discussions we had and his thoughtful advice and support. I would like to extend my deepest appreciation to Doctor Andrew Tomkins, for giving me the opportunity to discuss my research, for his invaluable advice for my life and my career, and for his support during a wonderful summer that I spent in his group during my internship at Google Research. Andrew has been as a mentor to me, and for that I feel very honored and grateful to him.

I am thankful to my Master thesis advisor, Professor Farhad Oroumchian, who introduced me to the field of information retrieval and first taught me how to do research. I owe him a big thanks for his tireless support during these many years of study.

I have many reasons to thank Doctor Ryen White. I worked with Ryen at Microsoft Research for another unforgettable summer internship. Ryen has been an essential component in my success, as I have gleaned a great deal of my own methodology from observing his unique perspective and vision in research. I am very thankful for his tremendous help and his continuous support on my study and career.

I have been blessed to receive a great deal of help from many collaborators, colleagues, and friends at UIUC. I would like to express my thanks to the members of the TIMan Group for many valuable discussions,

especially, Xuanhui Wang, Bin Tan, Xu Ling, Alexander Kotov, V.G. Vinod Vydiswaran, Duo Zhang, Hyun Duk Kim, Hongning Wang, Huizhong Duan, Kavita Ganesan, and Dae Hoon Park. I would also like to thank Majid Kazemian for the many fruitful discussions that we had.

This thesis would not have been possible without the emotional support and many light-hearted moments I shared with my good friends. Special thank goes out to my friend, Afsaneh Shirazi, for always being there for me through good times and bad. I also would like to express my sincere gratitude to Ghazaleh Hosseinabadi, who helped me immensely in getting through the difficult times during the last year of my PhD. And to all the other friends who made living in Urbana much so easier, I thank you as well. I hope the best for each of you in the coming years.

I want to express my great thanks to Google Inc. for granting me the Google PhD fellowship for two years, Yahoo! Inc. for the Yahoo! Key Scientific Challenge award, Facebook for the Grace Hopper Scholarship, and CS department at UIUC for Sohaib and Sarah Abbasi fellowship to support my PhD study.

I really do not know how to do justice in thanking my parents for their endless support, unconditional love, wholehearted encouragement, vital guidance, and for being such great role models. This work would not have been completed without their constant love, support and for their endurance in having me at such a long distance away. I am deeply thankful to my grandparents; I wish they could be here to see the day that I earn my PhD. They would have been so proud. I would like to thank them for believing in me, motivating me, and encouraging me in all aspects of my life, from the first grade up to my adult life. I am thankful beyond words to my sisters, Sara and Sahar, I cannot thank them enough for all they did for me; they have always believed in me even at the times when I did not believe in myself. Without their love and support, I would not have been the person I am today. This thesis is dedicated to my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the advent of Web, an explosive growth of online information, including Web pages, news articles, email messages, scientific literature, and information about all kinds of products on the Web, etc. are being generated across the globe at an unfathomable rate and covering countless topics. This dramatic growth in text information and the increasing number of ways people can utilize it has influenced our daily lives in fundamental and profound ways. This growth makes it more challenging to manage useful information effectively and efficiently for the users. Users are usually overwhelmed with the huge amount of information and have an urgent need for more powerful information retrieval systems. The widespread and casual use of search engines is one example of how transformed our lives have become in relation to text information. Search engines are to find relevant information from large amounts of texts and thus have now become essential tools in all aspects of our life; clearly, their effectiveness would directly affect our productivity and quality of life.

Although a variety of information needs can be served very successfully by current search engines, there are still a lot of queries that search engines cannot answer accurately which make users frustrated. For example, when a user does not have any particular pages in mind or does not know well about the topic to be searched, as is often the case in exploratory search and informational search [92], such short keyword queries are not always effective. When a query is so difficult that a large number of top-ranked documents are non-relevant to the user information need, a user would have to either reformulate the query or go far down on the ranked list to examine more documents, both may decrease the user satisfaction. As a result, improving the effectiveness of search results for such difficult queries would bring user satisfaction which is the ultimate goal of search engines.

The study of difficult queries has just started attracting attention recently due to the launch of the ROBUST track in the TREC conference [145, 146], which aims at studying the robustness of a retrieval model and developing effective methods for difficult queries. However, the most effective methods developed by the participants of ROBUST track tend to rely on external resources to perform query expansion, which has in some sense bypassed the difficulty of the problem because in reality, there are often no such external

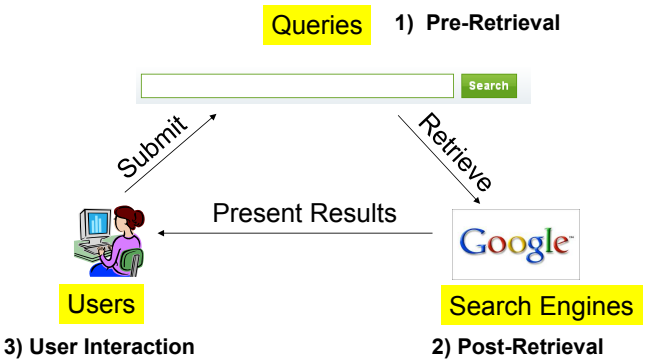

Figure 1.1: Three natural stages in interactive search process.

sources to exploit, or otherwise, the user would have directly gone to the external sources to find information. There has been some work on understanding why a query is difficult [12, 17, 52] or identifying difficult queries [146] or on predicating query performance [30, 54, 56, 61, 127, 160], but none of this work has systematically addressed how to improve the search accuracy for difficult queries from multiple perspectives.

In this dissertation, we systematically study how to improve the accuracy of search engines from different perspectives, naturally corresponding to different stages of an interactive search process. First, the user may not be able to formulate an effective query or he/she might not use the exact query terms that appear in relevant documents and thus would need support to bridge the vocabulary gap. This stage is called pre-retrieval stage. Second, for a difficult query, a user would have to browse through a long list of results before reaching the very first relevant document, and thus would benefit if the system can learn from the feedback information from top-ranked non-relevant documents. This stage is called post-retrieval stage. Third, When a query is difficult, a user is unlikely satisfied with only one interaction with the search engine; so there will be more interactions to get the desired results. Optimizing the utility from user perspective in the entire session of the user interaction with the system is called interactive retrieval stage. Figure 1.1 shows these stages in more details.

Each stage is described in more details in the followings:

Pre-retrieval stage: The accuracy of any search engine is mainly determined by the retrieval model that it uses. Language models have been widely used in text retrieval, leading to many state of the art retrieval models. A key challenge of language modeling approaches is how to estimate a robust language model from the very sparse information in a document or how to smooth a language model. Most of the retrieval models rely on exact matching of users' information need with a document. Since a user might not use the same exact vocabulary used in relevant documents and express his information need in many different ways, how to bridge the vocabulary gap is an important research problem that also helps improve the accuracy of difficult queries. Berger and Lafferty [8] proposed to assess the relevance of a document to a user's query by estimating the probability that the query would have been generated as a translation of the document. For estimating the translation probabilities, they proposed the idea of synthetic queries as their training data to learn the estimation probabilities. Some other study [62] uses (title, document) pairs as training data. These estimation methods are really inefficient and the coverage of query words is limited. In this dissertation, we propose methods to estimate the translation probabilities based on mutual information-based approach [67] and conditional-based approach [70] which are more efficient and accurate than the existing methods. Chapter 3 discusses it in more details.

Since the effectiveness of the translation language model would be mainly determined by the accuracy of the estimated translation probabilities, knowing whether one estimated translation language model is better than another is a key to the optimal behavior of the retrieval. No previous work has attempted to systematically analyze if a translation language model is optimal. To address this limitation and further analyze the optimality of a translation language model, we perform axiomatic analysis [38] which has been shown to be useful for diagnosing weakness of a retrieval model to gain insight about how to optimize the estimation of a translation language model [70]. We then propose five constraints that a reasonable translation language model should satisfy and check these constraints on the estimation translation language models. The results indicate that the mutual information-based approach does not satisfy all the constraints and the conditional-based estimation method better satisfies the constraints. Chapter 4 discusses it in more details.

With the prevalence of social media applications, an increasing number of internet users are actively publishing text information online. This influx provides a wealth of text information on those users. Ranking in social media poses different challenges than Web search ranking, one of which is that microblog messages are really short. As a result, the vocabulary mismatch problem is exacerbated in social media search, making it interesting to study how effective statistical translation language models are for improving social media search. In this dissertation, we study the standard translation language model for ranking short microblog

messages for Twitter [143] because it has gained huge popularity since the first day that was launched. More specifically, we study whether and how we can use statistical translation language model for ranking short tweet messages and reveal that translation language model not only helps to bridge the vocabulary gap but also improves the estimate of term frequency (TF) heuristics in information retrieval where the words appear only once in a tweet and there is no discrimination between the words. We further propose two ways to improve translation language model through leveraging hashtag information and adaptively setting the self-translation probability. Experimental results on Twitter data set show that our proposed methods are effective. Chapter 5 discusses it in more details.

Post-retrieval stage: When a query is very difficult that the search results are poor, getting feedback information from the user is a very helpful method to improve the retrieval accuracy and user experience. For example, if a user clicks on a document, we can use useful related terms in that document to reformulate the query (positive feedback). In a case where the query is very difficult that none of the top-ranked documents are relevant, we can learn from negative examples (i.e., documents that a user does not click on.) to perform negative feedback. There have been some studies on negative feedback [148, 149] where the authors studied different methods for negative feedback in both language models and vector space model and concluded that negative feedback for language modeling approach works better. One challenge in negative feedback is that negative documents tend to be distracting in different ways, thus as training example, negative documents are sparse. In this dissertation, we solve the problem of data sparseness in the language modeling framework. We propose an optimization framework in which we learn from a few top-ranked negative documents and search in the space of all candidate language models to build a more general negative language model. This general negative language model has been shown to have more power in pruning non-relevant documents outperforming the state of the art negative feedback methods significantly for difficult queries [69]. Chapter 6 discusses it in more details.

Interactive retrieval stage: When a query is difficult, the user is unlikely satisfied with only one interaction with the search engine; so there will be more interactions to get the desired results. When presenting the search results to the user, there is a tradeoff between presenting search results with the highest immediate utility to a user (but not necessarily most useful for collecting feedback information) and presenting search results with the best potential for collecting useful feedback information (but not necessarily the most useful documents from a user's perspective). Optimizing such a tradeoff (called exploration-exploitation tradeoff) is a key to the optimization of the overall utility of feedback to a user in the entire session of user inter-

action with the system and has not studied before. We frame this tradeoff as a problem of optimizing the diversification of search results where we propose a machine learning approach to adaptively optimizing the diversification of search results for each user query so as to optimize the overall utility in an entire session [68]. In this optimization framework, we also propose to model the patience of the user [71]. Evaluation results indicate that optimizing the exploration-exploitation tradeoff is important to optimize the whole utility over an entire session in an interactive search and optimizing such a tradeoff outperforms the traditional relevance feedback method. Chapter 7 discusses it in more details.

The contributions from this dissertation are summarized as below:

- We propose methods to estimate the translation probabilities based on mutual information and conditional context which are more efficient and accurate than the state of the art estimation methods.
- We perform axiomatic analysis which has been shown to be useful for diagnosing weakness of a retrieval model to gain insight about how to optimize the estimation of a translation language model. We then propose five constraints that a reasonable translation language model should satisfy and check these constraints on the proposed estimation translation language models.
- We study whether a direct application of standard translation language model would improve the performance for Twitter search and find that it improves the performance. Further analyses reveal that statistical translation language model can have two benefits; it helps to bridge the vocabulary gap and in case when there is no vocabulary gap it still helps by improving the estimate of term frequency especially for short tweets where the words appear only once and there is no discrimination between the words. We further propose two ways to improve translation language model through leveraging hashtag information and adaptively setting the self-translation parameter which show further improve the performance.
- We propose an optimization framework in which we learn from a few top-ranked negative documents and search in the space of all candidate language models to build a more general negative language model. This general negative language model has been shown to have more power in pruning non-relevant documents outperforming the state of the art negative feedback methods significantly for difficult queries.
- When presenting the search results to the user, there is a tradeoff between presenting search results with the highest immediate utility to a user and presenting search results with the best potential for collecting useful feedback information. Optimizing such a tradeoff (called exploration-exploitation

tradeoff) is a key to the optimization of the overall utility of feedback to a user in the entire session of the user interaction with the system. We frame this tradeoff as a problem of optimizing the diversification of search results where we propose a machine learning approach to adaptively optimizing the diversification of search results for each user query so as to optimize the overall utility in an entire session which outperforms the traditional relevance feedback.

1.1 Plan of This Thesis

The results of this thesis have been partially reported before in [67, 68, 69, 70, 71, 72]. The rest of this thesis is organized as follows:

- Chapter 2 discusses all the previous work related to this dissertation including retrieval models, semantic term matching, difficult queries, diversification and social networks.
- Chapters 3, 4 and 5 discuss about the pre-retrieval stage where in Chapter 3 we focus on proposing new efficient and accurate estimation methods, in Chapter 4 we focus on performing axiomatic analysis of translation language model for retrieval in order to gain insights about how to optimize the estimation of translation probabilities. Finally in Chapter 5, we study the standard translation language model for ranking short microblog messages for Twitter.
- Chapter 6 discusses the post-retrieval stage where an optimization framework in which we learn from a few top-ranked negative documents and search in the space of all candidate language models to build a more general negative language model is proposed.
- Chapter 7 discusses optimizing the whole interaction session with the user after post-retrieval stage to optimize the exploration-exploitation tradeoff which is a key to the optimization of the overall utility of feedback to a user in the entire session of user interaction with the system.
- And finally Chapter 8 concludes this dissertation and discusses a few directions for future work.

Chapter 2

Related Work

In this chapter, we review all the previous relevant work to this dissertation including retrieval models, semantic term matching, feedback techniques, difficult queries, diversification and social networks.

A basic information retrieval problem set up is as follows: We assume there exists a *document collection* $\mathcal{C} = \{d_1, \ldots, d_n\}$ where d_i is a text document. Given a *query* (user information need) q, the task of an information retrieval system is to return a *ranked list* of documents so that the documents which are ranked on the top are more relevant to the query than those ranked below of them. *Relevant documents* are what the user is looking for, i.e., they can be regarded as containing the expected answers to the query. *Ranking* is done by using a *retrieval function* S to score each document with respect to the query and then ranking all the documents based on their scores. In many retrieval models, the order of words in a query or in a document is ignored, thus both query and document would be a *bag of words*. The bag of words representation has been shown to perform quite well which is a popular representation in all search engines.

2.1 Retrieval Models

Many retrieval models such as vector space model [120, 122, 123], probabilistic models [43, 66, 79, 115, 103], probabilistic inference models [42, 46, 77, 110, 111, 142, 112, 154] and axiomatic retrieval framework [38, 39, 40, 41] have been proposed previously. In this section, all the major retrieval models are briefly reviewed.

2.1.1 Similarity-Based Models

In these models, it is assumed that the relevance of a document with respect to a query is correlated with the *similarity* between the query and the document. The vector space model [120, 122, 123] is the most well-known model for this type where a document and a query are represented as two term vectors in a high-dimensional term space. Each term defines an independent dimension. The value of each vector component is related to the weight of the corresponding term in the given text. Term weighting is usually assigned based on different heuristics, such as TF-IDF weighting (term frequency-inverse document frequency) and

document normalization. The relevance score between a query and a document can be computed based on the similarity between two corresponding vectors. The most commonly used similarity function is cosine similarity. The term weighting scheme is outside the relevance modeling framework, which makes it hard to control and explain the parameters. As a result, retrieval performance is very sensitive to the parameter setting.

One of the best performing vector space retrieval function is *pivoted normalization* retrieval function [132] which is as follows:

$$S(q,d) = \sum_{t \in q,d} \frac{1 + ln(1 + ln(c(t,d))}{(1-s) + s\frac{dl}{avdl}} . c(t,q) . ln\frac{N+1}{df(t)} \tag{2.1}$$

Where s is an empirical parameter (usually 0.2), and
$c(t,d)$ is the term's frequency in document
$c(t,q)$ is the term's frequency in query
N is the total number of documents in the collection
df is the number of documents that contain the term
dl is the document length, and
$avdl$ is the average document length.

2.1.2 Probabilistic Relevance Models

In probabilistic relevance model, given a query, a document is assumed to be either relevant or non-relevant, so the system has to rely on a probabilistic relevance model to estimate it. Such a retrieval strategy can be justified by the Probability Ranking Principle (PRP) [114]. Under this assumption, the PRP provides a justification for ranking documents in decreasing order of probability of relevance. The notion of relevance is then captured through the binary random relevance variable and a probabilistic model is defined to associate this variable with some probabilistic representation of documents and queries. Without relevance example, the estimation of parameters can be difficult; thus, heuristics may be needed to make a model useful for ad hoc retrieval. For example, an approximation of 2-Poisson mixture model has led to a quite effective retrieval function, BM25 [116]. The BM25 retrieval formula is as follows:

$$s(q,d) = \sum_{t \in q,d} (ln\frac{N - df(t) + 0.5}{df(t) + 0.5} \times \frac{(k_1+1) \times c(t,d)}{k_1((1-b) + b\frac{dl}{avdl}) + c(t,d)} \times \frac{(k_3+1) \times c(t,q)}{k_3 + c(t,q)}) \tag{2.2}$$

Where $k_1 \in [1.0, 2.0]$, b (usually 0.75) and $k_3 \in [0, 1000]$ are parameters and other variables have the same meaning as in the vector space retrieval formula.

2.1.3 Probabilistic Inference Model

In a probabilistic inference model [42, 46, 77, 110, 111, 112, 142, 154], the uncertainty whether a document is relevant to a query is modeled by the uncertainty associated with inferring the query from the document. Various definitions of "inferring a query from a document" lead to different inference models. The probabilistic inference model must rely on further assumption about the representation of documents and queries in order to obtain an *operational* retrieval formula. The choice of such representations is in a way outside the model, so there is little guidance on how to choose or how to improve a representation. The inference network model is also based on probabilistic inference [142] which is essentially a bayesian belief network that models the dependency between the satisfaction of a query and the observation of documents. The estimation of relevance is based on the computation of the conditional probability that the query is satisfied given the observed document.

In general, the probabilistic inference models address the issue of relevance in a very general way. Their lack of commitment to specific assumptions in these general models has helped to maintain their generality as retrieval models but they generally provide little guidance on how to refine the general notion of relevance.

2.1.4 Axiomatic Retrieval Framework

The three retrieval model categories described have some parameters that need to be tuned in order to obtain optimal retrieval performance. If the retrieval parameter is not set optimally, their performance can be very poor. In addition, these models eventually lead to some retrieval function that implements TF-IDF weighting and document length normalization. These weighting heuristics must be implemented in some special functional form in order for retrieval function to perform well.

In [38], the three major retrieval heuristics (i.e., TF, IDF and length normalization) are formally described with well-defined constraints on retrieval functions. It is also shown that the empirical performance of a retrieval function is correlated with whether they satisfy the constraints. Since all reasonable retrieval functions must satisfy such constraints, these constraints can be regarded as axioms for a retrieval function. These axioms can be used to guide us in finding an effective retrieval function by satisfying all the constraints. In [39, 40, 41] an axiomatic retrieval framework has been leveraged to derive some interesting new retrieval functions that are more robust than existing retrieval functions. Although, the framework offers a novel way to explore different retrieval models, the framework does not provide guidance on what candidate functions to explore. Thus, it is still needed to rely on existing retrieval models to suggest a tractable search space. As an extension of previous work, we propose new constraints for statistical translation language model.

2.1.5 Probabilistic Distance Retrieval Model

The major problem with query likelihood retrieval model is that it cannot easily handle relevance or pseudo-relevance feedback. In these models, a document is represented with a document language model and a query is represented with a query language model. A document is scored based on the distance between the corresponding language models using some probabilistic distance measure such as Kullback-Leibler (KL) divergence [78]. This model is a generalization of the query likelihood retrieval model and would score a document d w.r.t query q based on the negative Kullback-Leibler divergence between the query language model θ_q and the document language model θ_d:

$$S(d,q) = -D(\theta_q||\theta_d) = -\sum_{w \in V} p(w|\theta_q) \log \frac{p(w|\theta_q)}{p(w|\theta_d)} \tag{2.3}$$

where V is the words in the vocabulary.

Clearly, the two main tasks are to estimate the query language model θ_q and the document language model θ_d. The document language model θ_d is usually smoothed using Dirichlet prior smoothing which is an effective smoothing method [165].

The query model intuitively captures what the user is interested in, thus would affect retrieval accuracy significantly. The query language model, is often estimated (in case of no feedback) based on $p(w|\theta_q) = \frac{c(w,q)}{|q|}$, where $c(w,q)$ is the count of word w in query q and $|q|$ is the total number of words in the query. Such a model is not very discriminative because a query is typically extremely short. When there is feedback information, the information would be used to improve our estimate of query language model, θ_q.

Feedback techniques have proven to be very effective for improving retrieval performance (e.g. [5, 51, 68, 81, 115, 118, 121, 129, 155, 164]). After a search engine presents some results to a user, sometimes the user is willing to provide some feedback on the relevance status of the results (relevant or non-relevant). In such a case, the retrieval system can learn from the examples of relevant or non-relevant documents provided by the user to improve the search results. This is called *relevance feedback*. When the user is not willing to make any judgments, the system still can perform feedback by assuming some top-ranked documents to be relevant. This is called *pseudo-relevance feedback*. A third kind of feedback is to use user interactions (e.g., past queries, click-through) to infer user's intent and improve search results. This is called *implicit feedback* [73]. The fourth kind of feedback is active feedback. Active feedback is essentially an application of active learning which has been extensively studied in machine learning (e.g. [23, 119]). Active learning has been applied to text categorization (e.g. [86, 93, 140]) and information filtering [167]. Recently, it has also been applied to ad hoc retrieval [59] and relevance feedback [130, 158, 157].

Most feedback methods rely on positive documents, i.e., documents that are judged as relevant to provide useful related terms for query expansion. In contrast, negative (non-relevant) documents have not been found to be very useful. However, there have been some attempts to exploit non-relevant documents; query zone [133] appears to be the only major heuristic proposed to effectively exploit non-relevant information for document routing tasks. It showed that using non-relevant documents that are close to the original query is more effective than using all non-relevant documents in the collection. However, this problem was studied for document routing tasks and a lot of relevant documents are used. Also, the work in [109] exploits high-scoring documents outside of top N documents (called pseudo-irrelevant documents) to improve the performance of pseudo-relevance feedback. The work in [148] and later extension to that [149] are the first studies of negative relevance feedback that exploit only the top non-relevant documents to improve the ranking of documents and they start with non-relevant documents close to a query to study how to use this negative information optimally in ad hoc retrieval. Our work in this dissertation defines an important concept called *generalization of a language model* and we propose an optimization framework based on this concept to more aggressively (but carefully) prune non-relevant documents, leading to a more effective negative feedback method. Since we define an optimization framework for negative feedback, our work is also related to previous optimization techniques for pseudo-relevance feedback [24, 25, 32]. Our work is similar to all this work since we also define an optimization framework for term selection. However, it differs in that 1) our optimization framework is defined based on generalizing a negative language model, an important concept that none of the previous work considered; 2) our optimization framework is for negative feedback not for positive feedback, thus helps improving difficult queries.

2.2 Semantic Term Matching

Many existing retrieval functions assume that relevance score can be computed solely based on the exact matching of query terms. But in reality, this assumption does not hold because it is unlikely that a user would use a query term that is exactly the same one used in relevant documents. As a result, many studies have tried to bridge the vocabulary gap between documents and queries mostly based on either the co-occurrence thesaurus [7, 63, 82, 102, 105, 128, 134, 155] or hand-crafted thesaurus [88, 147]. Some other studies have considered to combine both approaches [14, 91]. Also, the query expansion work in [27] used a term dependency graph in which word co-occurrence was one of the several dependency types. In this dissertation, we consider word co-occurrence relationship based on mutual information and incorporate it into statistical translation language model in a more principled way. Although it is applicable to exploit

both types of thesauri in statistical translation language model, in this dissertation, we focus on the use of co-occurrence-based thesaurus and leave other possibilities as future work.

In language modeling framework, statistical translation language model has been introduced to incorporate term relationship into language modeling approaches. Statistical translation models were originally studied in machine translation with the goal of automatically translating sentences between different languages (e.g., French and English) [10] where authors proposed five different translation models. The simplest model (i.e., IBM 1) [10] ignores position information when learning word-to-word translation probabilities. This model has been adopted in information retrieval by Berger and Lafferty [8] to incorporate term relationship into language modeling approaches. Theoretically, the translation language model provides a principled way to support semantic matching of related words. To train translation language models, they synthetically generated (query, document) pairs. An alternative way of estimating the translation model is based on document titles [62]. In this work, the authors proposed to use (title, document) pairs as training data. These estimation methods are inefficient and the coverage of query words is low due to sampling. In this dissertation we systematically analyze the optimality of the translation language model and propose estimation methods that are more efficient and accurate.

Translation language models have been naturally used in cross-lingual information retrieval domain [98, 156]. For example, Nie et al. [98] used parallel corpus as training data to learn translation models. The work by Lavrenko et al. [80] has adapted the relevance model in two different ways based on KL-divergence retrieval models to perform cross-lingual information retrieval. The cluster-based query likelihood proposed in [76] can be regarded as a form of a translation model where the whole document is translated into the query. Recently, translation models have been applied in many applications including question answering, sentence retrieval and tracking information flow [95, 97, 159]. For example, Xue et al [159] has applied translation model on question-answer archives where question and answer pairs are used to train the translation model. In contrary to all these studies, we studied statistical translation language model in *ad hoc retrieval* context.

2.3 Difficult Queries

The study of difficult queries has attracted much attention recently especially with the launch of ROBUST track in TREC conference which aims at studying the robustness of retrieval models [145, 146]. However, the most effective retrieval models that are developed by ROBUST participants relied on external resources (mostly Web) to perform query expansion which has bypassed the difficulty of the problem in reality. Because there are often no such external resources to exploit and indeed the Web resources would not help improve

the search accuracy for difficult queries on the Web itself. In this dissertation, we aim at exploiting negative feedback information in the target collection through an optimization framework.

Queries are difficult due to various reasons. Understanding why a query is difficult and categorizing difficult queries into different causes are critical to developing effective retrieval strategies. There has been some work on understanding why a query is difficult [12, 17, 52, 126]. Savoy [126] analyzed the reasons of difficult topics from a query perspective. Harman and Buckley [12, 52] conducted the analysis from a retrieval engine perspective. However, their focus is on identifying the causes, but do not propose methods to address the problems.

More recent work on difficult queries are on predicting query difficulty [30, 54, 56, 61, 127, 160]. In [30], clarity score is defined to measure the difference between a query language model and a collection language model. They found that the smaller the score is, the more difficult the query is. More features are designed and analyzed in [17, 160] to estimate query difficulty. All this work predict whether a query is difficult but do not try to predict the causes of why a query is difficult.

Most of previous work on difficult queries has not addressed the important question of how to improve search accuracy for difficult queries. In this dissertation, we systematically study this problem by proposing specific techniques along different perspectives.

2.4 Diversification and Interaction

Since we optimize the exploration-exploitation tradeoff, in this section, we review the related work. Relevance feedback methods that are discussed in the previous section, only consider "exploitation", and our study is orthogonal to optimization of these relevance feedback algorithms. We used a mixture model feedback method [164] in this dissertation, but our idea is general and can potentially work for other feedback methods too. All active feedback methods emphasize on "exploration" only, and are also orthogonal to our study. We adopted the diversification strategy for active feedback, but the proposed methods are also potentially applicable to other methods for active feedback.

Multi-arm bandit methods that seek to find the optimal tradeoff between exploration and exploitation have been studied extensively (e.g., [99, 100, 108, 162]). Existing solutions to the standard bandit problem assume a fixed set of arms with no delayed feedback. However, recently, the work in [2] has considered such a tradeoff to maximize total clicks on a web content. Also, reinforcement learning has been used for solving sequential decision making problems, which assumes there exists an agent interacting with the environment with the goal of maximizing the total reward. There have been extensive applications adopting reinforcement

learning (e.g. [1, 65, 85, 131, 166]). Our work in this dissertation is similar to all this work in that we also maximize the total utility by optimizing the tradeoff between exploration and exploitation, but we explore a new application in optimizing interactive relevance feedback.

Our work in this dissertation is also related to previous work in diversifying search results. Goffman [48] recognized that the relevance of a document must be determined w.r.t documents appearing before it. Several researchers have been working on methods to eliminate the redundancy in the result sets (e.g. [9, 16, 20, 163, 169]) by sequentially selecting documents that are relevant to the query but dissimilar to documents ranked above them or by introducing an evaluation framework that rewards novelty and diversity and penalizes redundancy [21]. Some other approaches [18, 161] maximize diversity (topic coverage) among a set of documents without regard for redundancy [18] or by learning the importance of individual words and then selecting the optimal set of results that cover the largest number of words [161]. Some other work on diversification make use of taxonomy for classifying queries and documents and create a diverse set of results according to this taxonomy [3, 144, 170]. Generating *related* queries and taking the results from each of them for re-ranking purposes [106] is another way for diversification. In addition, some other work, consider diversity in the area of spatial information retrieval [137] or in image retrieval [101, 124] or for query diversification [22]. All these studies consider diversity in terms of providing a complete picture of different aspects of the query. However, we consider diversity in terms of providing more information for optimizing the utility of relevance feedback over an interactive retrieval session. We used the method in [163] however, our approach can be applied to any diversification method where there is a novelty parameter.

Learning to rank has recently attracted much attention (e.g. [13, 15, 57, 64, 107]). Most of the work in this line has not considered diversity. Logistic regression [53] is widely used to learn a retrieval function to rank documents directly [19, 28, 45, 47, 157, 168]. Our work uses logistic regression to learn the diversity parameter.

In [108] an online learning algorithm that directly learns a diverse ranking of documents based on user's clicking behavior to minimize abandonment (maximizing clickthrough) for a single query is proposed. While abandonment is minimized, their approach cannot generalize to new queries. The difference between our work and theirs is that they optimize the tradeoff for multiple users and a single query, while we optimize such a tradeoff for a single user over multiple interaction cycles (i.e., over multiple pages).

Interactive search is another related line to our work [44, 85]. The author in [44] has proposed the holistic model to determine what is the best next action the system should perform when considering past interactions. However, none of this work has provided a technique to optimize the exploration-exploitation tradeoff.

2.5 Social Networks

Our work is also related to social network analysis which has been a hot topic for quite some time. Many techniques have been proposed to discover communities [6, 74], model the evolution of the graph [83], and understand the diffusion of social networks [50, 84]. The work in [94] defines a topic model on the network structure. Our work is similar to all in that we also consider the problem in social network, however our problem setup is different in that we rank the short text messages for the search problem in social media applications.

More specifically on ranking in social media, the work in [152], extends PageRank to Twitter which make use of follow relationships in the Twitter graph as well as topical similarity to find the influential users for various topics. Then the work in [151] extends it by examining the semantics of follow links and proposing retweet links as an additional source of information. In addition, the work in [125] improves the ranking accuracy for the "Twitter-like" postings in forums with a comparison-based mechanisms. Tunkrank [141] proposes a ranking mechanism to identify the most influential Twitter users. The notion of influential is described as the expected number of users who will read a tweet from them. In addition, as described in [33], Twitter users are ranked with different criteria such as the number of followers, average content spread per tweet, average conversation activity per tweet and so on. Our work is similar to all this work in a sense that our problem is also a ranking problem; however we specifically consider to rank the short tweets by relying on the text content rather than considering the network structure.

A recent work [35] considers modeling temporal information in Twitter where time is considered as a factor in the retrieval model. In another work [34], hashtag retrieval is considered where the author proposes relevance feedback based on hashtags. Our work is similar to this work since we consider hashtags to help improve the performance in the statistical translation language model framework. However, our methods improve the ranking of Twitter search which is more general than hashtag retrieval.

Other Related Work: The document term frequency (TF), which dates back to Luhn's pioneer work on automatic indexing [89] has been playing a critical role in modern information retrieval models [4, 117, 132, 165]. It is widely recognized that linear scaling in term frequency puts too much weight on repeated occurrences of a term, as a result normalization is needed [132]. Since we consider ranking in Twitter and tweets are really short, most words appear only once in the tweets. Thus, term frequency discrimination is very important in Twitter search. We study statistical translation language model to discriminate between the words by getting support from other semantically related words.

Chapter 3

Pre-Retrieval: Statistical Translation Language Model for Information Retrieval

In this chapter, we talk about the pre-retrieval stage where the user might not use the same exact words in documents and the question is how to bridge the vocabulary gap and return relevant documents for users.

As a principled approach to capturing semantic word relations, statistical translation language models have been proposed for information retrieval to reduce the gap between documents and queries [8, 62]. Based on statistical machine translation [10], the basic idea of translation language models is to estimate the likelihood of translating a document to a query. Since a term has certain probability to be translated into a different term, translation language models can alleviate the vocabulary gap problem in a natural manner. As a result, translation language models have been successfully applied to different tasks such as cross-lingual information retrieval [80, 98, 156], question answering [159], sentence retrieval [97], and tracking information flow [95].

A main challenge in applying translation models to ad hoc information retrieval is to estimate a translation model without training data. Existing work has relied on training on synthetic queries generated based on a document collection [8]. However, this method is computationally expensive and does not have a good coverage of query words. In this chapter, we propose an alternative way to estimate a translation model based on normalized mutual information between words, which is less computationally expensive and has better coverage of query words than the synthetic query method of estimation. We also propose to regularize estimated translation probabilities to ensure sufficient probability mass for self-translation probabilities. Experimental results show that the proposed mutual information-based estimation method is not only more efficient, but also more effective than the synthetic query-based method, and it can be combined with pseudo-relevance feedback to further improve retrieval accuracy. The results also show that the proposed regularization strategy is effective and can improve retrieval accuracy for both synthetic query-based estimation and mutual information-based estimation.

In the next sections, we first review basic language modeling approach and translation language model.

3.1 Statistical Language Models

A statistical language model is a probability distribution over word sequences. It gives any sequence of words a different probability. Language modeling approaches received considerable attentions recently [103]. Given a language model, we can sample word sequences according to the distribution to obtain a text sample. Such a model can be used to generate a text. As a result, a language model is also called a generative model for text.

The simplest language model is the *unigram language model* in which a word sequence is generated by generating each word *independently*. As a result, the probability of a sequence of words would be equal to the product of the probability of each word.

The basic idea of a language model which is often called *query likelihood* scoring method can be described as follows [103]:

We assume that a query q is generated by a probabilistic model based on a document d. Given a query $q = q_1, q_2, \ldots, q_m$, and a document d, we are interested in estimating $p(d|q)$, i.e. the probability that document d has been used to generate query q. By applying Bayes' formula, we have:

$$p(d|q) \propto p(q|d)p(d) \tag{3.1}$$

$p(d)$ on the right hand side of the above formula is our *prior* belief that document d is relevant to any query. $p(q|d)$ is the query likelihood for the given document d, which intuitively measures how well document d matches query q. $p(d)$ is often assumed to be uniform and thus can be ignored for ranking documents. Further assuming that each query word is generated independently, i.e., assuming a multinomial language model, we would generate a sequence of words by generating each word independently, we can rewrite the above formula as (in the form of log likelihood):

$$\log p(d|q) \stackrel{\text{rank}}{=} \sum_{w \in V} c(w, q). \log p(w|d) \tag{3.2}$$

where $\stackrel{\text{rank}}{=}$ means equivalence for the purpose of ranking documents, $c(w, q)$ is count of word w in query q, and V is the vocabulary set. The challenging part is to estimate a document model $p(w|d)$. Based on multinomial distribution, the simplest way to estimate $p(w|d)$ is the *maximum likelihood estimator*:

$$p_{ml}(w|d) = \frac{c(w, d)}{\sum_{w'} c(w', d)} \tag{3.3}$$

Where $c(w, d)$ is count of word w in document d.

Due to the data sparseness problem (since a document is a very small sample for the model), maximum likelihood estimator underestimates the probability of unseen words in a document. *Smoothing* techniques address this problem by assigning non-zero probabilities to the unseen words and thus improving the accuracy of probability estimation. Specifically, smoothing is to discount the probabilities of words seen in the text and then assign extra probability mass to the unseen words according to some fallback model. Usually, collection language model is used as fallback model [165]. Two commonly used methods are Jelinek-Mercer and Dirichlet prior smoothing methods:

Jelinek-Mercer Method (JM Smoothing): This is a linear interpolation of maximum likelihood model with the collection model, using λ as a coefficient weight.

$$p(w|d) = (1 - \lambda)p_{ml}(w|d) + \lambda p(w|\mathcal{C}) \tag{3.4}$$

Where $p(w|\mathcal{C})$ is probability of word w in collection $\mathcal{C}$.

Bayesian Smoothing using Dirichlet Prior (Dirichlet Prior Smoothing): Since the conjugate prior of a multinomial distribution is the Dirichlet distribution, we can specify a Dirichlet prior distribution parameterized as

$$(\mu p(w_1|\mathcal{C}), \mu p(w_2|\mathcal{C}), \ldots, \mu p(w_n|\mathcal{C}))$$

where μ is a parameter. The estimated document model based on the posterior mean is then:

$$p(w|d) = \frac{|d|}{|d| + \mu} p_{ml}(w|d) + \frac{\mu}{|d| + \mu} p(w|\mathcal{C}) \tag{3.5}$$

The challenging part is to estimate a document model $p(w|d)$. The simplest way to estimate $p(w|d)$ is the maximum likelihood estimator. Another interesting way of estimating $p(w|d)$ introduced by Berger and Lafferty [8] is based on statistical machine translation [10]. In order to assess the relevance of a document to a user's query, they have estimated the probability that the query would have been generated as a translation of the document. In other words, they allow the query likelihood to be computed based on a *translation model* of form $p(w|u)$, which is the probability that word u is semantically translated to word w.

To put it more formally, in their model, the query likelihood can be calculated by using the following "translation document model":

$$p(w|d) = \sum_{u \in d} p(w|u)p(u|d) \tag{3.6}$$

where $p(w|u)$ is the probability of "translating" word u into word w and it allows us to score a document by counting the matches between a query word and semantically related words in the document. If $p(w|u)$ only allows a word to be translated into itself, the simple exact matching query likelihood would be achieved. However, $p(w|u)$ would in general allow us to translate u into other semantically related words with non-zero probabilities, thus achieving "semantic smoothing" of the document language model.

3.2 Smoothing for Translation Language Model

In this section, we consider statistical translation language model when combined with two basic smoothing methods described in section 3.1.

The basic component in the translation language model is $p(w|d) = \sum_{u \in d} p(w|u)p(u|d)$ which can be used to replace $p_{ml}(w|d)$ in all basic language model approaches. This will give us **1)** translation language model with Dirichlet prior smoothing and **2)** translation language model with Jelinek-Mercer smoothing. When we replace $p_{ml}(w|d)$ with $p(w|d) = \sum_{u \in d} p(u|d)p(w|u)$ in equation 3.5, we have the following:

$$p(w|d) = \frac{|d|}{|d| + \mu}[\sum_{u \in d} p(u|d) \cdot p(w|u)] + \frac{\mu}{|d| + \mu} p(w|C) \tag{3.7}$$

And when it is replaced with p_{ml} in equation 3.4, we have the following:

$$p(w|d) = (1 - \lambda)[\sum_{u \in d} p(u|d) \cdot p(w|u)] + \lambda p(w|C) \tag{3.8}$$

Equations 3.7 and 3.8 give us Dirichlet prior smoothing and Jelinek-Mercer (JM) smoothing with translation language model, respectively.

3.3 Estimation of Translation Language Model

The key part for translation language model is to learn the word-to-word translation probability, $p(w|u)$. It is clear that the performance of the proposed smoothed translation language model depends on the quality of the word-to-word translation probabilities. In the scenario of statistical machine translation [10], a parallel corpus of two languages is often assumed to be available, and the Expectation-Maximization (EM) algorithm [31] can be used to estimate a translation model.

The authors in [10] applied the EM algorithm on bilingual training data. In their simplest model, i.e.,

IBM model 1, the parallel corpus consisting of English and French sentence pairs is given,

$$S = (\mathbf{e}_1, \mathbf{f}_1), (\mathbf{e}_2, \mathbf{f}_2), \ldots, (\mathbf{e}_N, \mathbf{f}_N)$$

(Note that $\mathbf{e}$ refers to an English sentence while e is an English word. The same applies to $\mathbf{f}$ and f respectively.) Then, the probability of translating an English word e to French word f is calculated as follows:

$$p(f|e) = \lambda_e^{-1} \sum_{i=1}^{N} c(f|e; \mathbf{e}_i, \mathbf{f}_i) \tag{3.9}$$

where

$$c(f|e; \mathbf{f}, \mathbf{e}) = \frac{p(f|e)}{p(f|e_1) + p(f|e_2) + \ldots + p(f|e_l)} c(f; \mathbf{f}_i) c(e; \mathbf{e}_i) \tag{3.10}$$

where l is the length of the English sentence and $c(f; \mathbf{f}_i)$ is the count of French word f appearing in the French sentence $\mathbf{f}_i$, similarly, $c(e; \mathbf{e}_i)$ is count of the English word e in the English sentence $\mathbf{e}_i$. λ_e^{-1} is a normalization factor which makes the sum of translation probabilities for English word e to 1 and is estimated as follows:

$$\lambda_e = \sum_{f} \sum_{i=1}^{N} c(f|e; \mathbf{f}_i, \mathbf{e}_i) \tag{3.11}$$

EM algorithm is used to find the updated probabilities for $p(f|e)$.

In order to gain word-to-word probabilities in monolingual scenario, ideally, we should have a sample of queries and relevant documents, but since we do not often have, Berger and Lafferty [8] use the idea of *synthetic queries* as their training data. The idea is to take a document and synthesize a query to which the document would be relevant. There are different methods for synthesizing a query from a document. One way would be to sample words uniformly from the document, but this method would generate queries containing a number of common words such as *the*, *of*, etc. Preferable method would be a sampling algorithm biased in favor of words which distinguish the document from other documents.

In order to select words which are representative of a document, for each document $\mathbf{d} \in \mathcal{C}$, they compute the mutual information statistics [60] for each of its words according to:

$$I(w, \mathbf{d}) = p(w, d) \log \frac{p(w|d)}{p(w|\mathcal{C})} \tag{3.12}$$

where $p(w|d)$ is the probability of word w in document d, and $p(w|\mathcal{C})$ is the probability of word w in the collection. Their proposed algorithm for generating synthetic queries is shown in Figure 3.1, where synthetic

queries are sampled based on normalized mutual information $\tilde{I}$, and the Poisson parameter λ is set to 15. The resulting $(\mathbf{d}, \mathbf{q})$ of documents and synthetic queries are used to estimate the probabilities with the EM algorithm.

```
1.  Begin
2.    Do for each document d ∈ C
3.      Do for x = 1 to 5
4.      Begin
5.        Select a length m for this query according to Poisson distribution
6.        Do for i = 1 to m
7.          Select the next query word by sampling the scaled distribution: q_i ~ Ĩ
8.        Record (d, q)
9.      End
10.   End
```

Figure 3.1: A Sampling method for synthetic queries.

Although generating synthetic queries is a reasonable way to estimate the translation probabilities, this method has two deficiencies: (1) it is inefficient; (2) there is no guarantee that a query word is covered. In the next section, we propose a mutual information-based estimation which is more efficient than this method and has a better word coverage.

3.4 Estimation of Translation Language Model based on Mutual Information

In this section, a more efficient way to estimate translation probabilities is proposed which can have a better coverage of query words than the existing method discussed in the previous section.

3.4.1 Mutual Information-Based Approach

Mutual information [113] is a good measure to assess how two words are related. In our method, for each word in the collection, we compute all words which have high mutual information scores with it and normalize the computed mutual information scores as follows:

First, we compute the mutual information scores for each pair of two words w and u in the collection. Informally, mutual information compares the probability of observing w and u *together* (the joint probability)

with the probabilities of observing w and u *independently*. The mutual information between words w and u are calculated as follows:

$$I(w;u) = \sum_{X_w=0,1} \sum_{X_u=0,1} p(X_w, X_u) \log \frac{p(X_w, X_u)}{p(X_w)p(X_u)} \tag{3.13}$$

where X_u and X_w are binary variables indicating whether u or w is present or absent.

The probabilities are estimated as follows:

$$\begin{aligned}
p(X_w = 1) &= \frac{c(X_w = 1)}{N} \\
p(X_w = 0) &= 1 - p(X_w = 1) \\
p(X_u = 1) &= \frac{c(X_u = 1)}{N} \\
p(X_u = 0) &= 1 - p(X_u = 1) \\
p(X_w = 1, X_u = 1) &= \frac{c(X_w = 1, X_u = 1)}{N} \\
p(X_w = 1, X_u = 0) &= \frac{(c(X_w = 1) - c(X_w = 1, X_u = 1))}{N} \\
p(X_w = 0, X_u = 1) &= \frac{(c(X_u = 1) - c(X_w = 1, X_u = 1))}{N} \\
p(X_w = 0, X_u = 0) &= 1 - p(X_w = 0, X_u = 1) \\
&\quad -p(X_w = 1, X_u = 0) - p(X_w = 1, X_u = 1)
\end{aligned}$$

where $c(X_w = 1)$ and $c(X_u = 1)$ are the numbers of documents containing word w and u, respectively, $c(X_w = 1, X_u = 1)$ is the number of documents that contain both w and u, and N in the total number of documents in the collection.

We then normalize the mutual information score to obtain a translation probability:

$$p_{mi}(w|u) = \frac{I(w;u)}{\sum_{w'} I(w';u)} \tag{3.14}$$

$p_{mi}(w|u)$ gives us the probability of translating word u to another word w; intuitively, the probability would be higher if the two words tend to co-occur with each other.

3.4.2 Optimizing Self-Translation Probability

The approaches described in sections 3.3 and 3.4 might underestimate the self-translation probabilities, i.e., it is possible that $p(w|u) > p(w|w)$. This may lead to non-optimal retrieval performance because it is possible that a document that matches a query word exactly ($p(w|w)$) gets less score contribution from matching the query word exactly than a document that "matches" a query word through translation ($p(w|u)$). To overcome this bias, a parameter α is introduced to control the effect of self-translation. This is a general method that can be applied to adjust the estimated probabilities for any given estimation method.

$$p(w|u) = \begin{cases} \alpha + (1-\alpha)p_t(u|u) & w = u \\ (1-\alpha)p_t(w|u) & w \neq u \end{cases} \tag{3.15}$$

and $p_t(w|u)$ is estimated with any estimation methods such as mutual information or synthetic queries. α is a parameter that controls the effect of self-translation probability and when we set $\alpha = 1$, we recover the basic query likelihood method.

The "regularized" translation model $p(w|u)$ can then be used in Equations 3.7 and 3.8 to rank documents.

3.4.3 Translation Language Model with Feedback

Feedback techniques have been shown to improve retrieval accuracy substantially[81, 115, 164]. A natural question with translation language model is whether translation model can benefit from feedback techniques. In this section, we use pseudo-relevance feedback to expand our query model [164] and then score the expanded query model with translation language model based on the negative cross entropy of the expanded query language model and the translation document model (also equivalent to scoring based on negative KL-divergence):

$$\sum_{p(w|\theta_q)>0} p(w|\theta_q).\log p(w|d) \tag{3.16}$$

where $p(w|\theta_q)$ is the query model generated by pseudo-relevance feedback and $p(w|d)$ is a smoothed translation model and can be computed using either of equations 3.7 or 3.8.

3.5 Experiments

3.5.1 Data Set

The experiments in this section use four main document collections: (1) news articles (AP90) with TREC topics 51-100 and 78,321 articles. (2) San Jose Mercury News (SJMN) articles with TREC topics 51-100

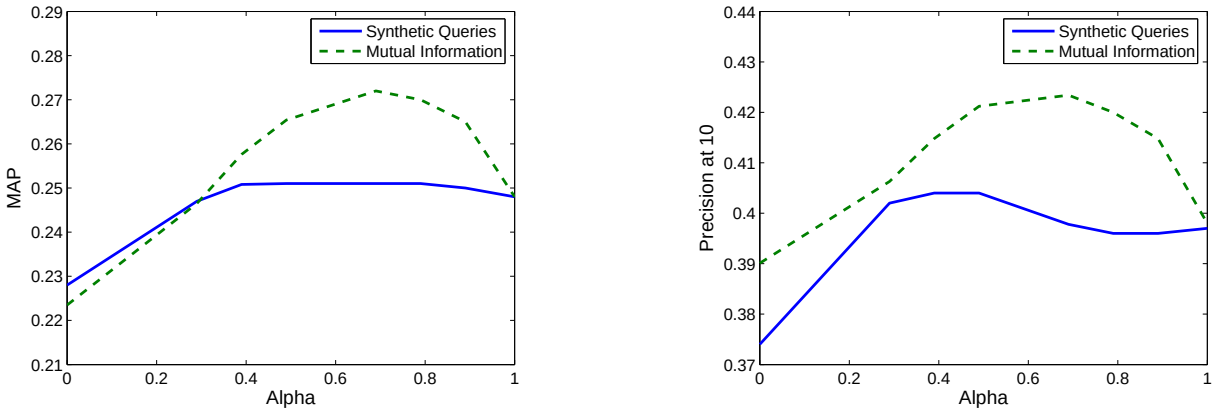

Figure 3.2: Comparison of mutual information and synthetic queries according to MAP (Left) and Precision at 10 (right). (Both are according to Dirichlet prior smoothing).

and 90,250 articles (3) ad hoc data in TREC7 with topics 351-400 and 528,155 articles and (4) TREC8 with topics 401-450 and 528,155 articles.

In the experiments, we only use title of the queries. As for preprocessing, we do stemming using Porter stemmer [104] and stop word removal. All experiments are done using the Lemur toolkit [1]. The performance is measured using two standard measures: MAP(Mean Average Precision) and Precision @10 (precision at 10).

The optimal value for Dirichlet prior smoothing for baseline is 1000 for all data sets and optimal value for JM smoothing for baseline method is gained when coefficient is set to 0.5 for AP90 data set and 0.3 for the rest of data sets.

The methods used for experiments in the following sections are: BL (baseline), i.e., either Dirichlet prior smoothing or JM smoothing (Equations 3.4 or 3.5), MI (translation language model with mutual information[2] for word-to-word translation probabilities), SYN (translation language model with synthetic queries), fb (pseudo-relevance feedback on baseline) and fb+TM(pseudo-relevance feedback combined with translation language model using mutual information).

3.5.2 Comparing Synthetic Queries with Mutual Information

We first look into the question whether mutual information (MI) can be an alternative way of estimating translation model. Table 3.1 shows the results for both SYN and MI methods with both Dirichlet prior smoothing and JM smoothing, respectively. The results indicate that MI method is better able to capture

[1] http://www.lemurproject.org/

[2] We use mutual information throughout the dissertation for simplicity but we mean the normalized mutual information described in section 3.4.

Table 3.1: Performance of translation language model with synthetic queries and mutual information estimation according to Dirichlet prior smoothing (left) and JM smoothing (right), * means improvements over SYN are statistically significant with Wilcoxon signed-rank test. We only show the significance tests for MAP measure.

Data	MAP		Precision @10	
	MI	SYN	MI	SYN
AP-90	0.272*	0.251	0.423	0.404
SJMN	0.2*	0.195	0.28	0.266

Data	MAP		Precision @10	
	MI	SYN	MI	SYN
AP-90	0.264*	0.25	0.381	0.357
SJMN	0.197*	0.189	0.252	0.267

Table 3.2: Sample word translation probabilities using synthetic queries (left) and mutual information (right). Note that words are stemmed.

w=everest	
q	$p(q\|w)$
everest	0.079
climber	0.042
climb	0.0365
mountain	0.0359
mount	0.033
reach	0.0312
expedit	0.0314
summit	0.0253
whittak	0.016
peak	0.0149

w=everest	
q	$p(q\|w)$
everest	0.1051
climber	0.0423
mount	0.0339
028	0.0308
expedit	0.0303
peak	0.0155
himalaya	0.01532
nepal	0.015
sherpa	0.01431
hillari	0.01431

word relatedness. Indeed, statistical significance tests indicate that the difference between MI and SYN is statistically significant. In addition, estimating translation probabilities by mutual information for all data sets is more efficient than learning translation probabilities by synthetic queries. Table 3.2 shows a document word together with ten most probable query words that it will translate to by both synthetic queries and mutual information estimation methods. The table shows that the related words for word "everest" in case of mutual information are more specific than for words learned via synthetic queries.

Figure 3.2 shows the sensitivity of mutual information and synthetic queries to α parameter according to MAP measure (left) and Precision@ 10 (right). The difference indeed makes clearer that mutual information works better than synthetic queries. (Our results for synthetic queries are comparable to those reported in [8].)

According to these results, we can conclude that mutual information works better than synthetic queries and it is also more efficient.

Because of the high computational complexity of synthetic queries, we cannot compare mutual information with it on larger collections, but later we will further experiment with mutual information on larger collections.

Table 3.3: Performance of translation language model on different datasets with Dirichlet prior smoothing (left) and JM smoothing (right), * means improvements over baseline are statistically significant with Wilcoxon signed-rank test. We only show the significance tests for MAP measure.

Data	MAP		Precision @10	
	BL	MI	BL	MI
AP-90	0.248	0.272*	0.398	0.423
SJMN	0.195	0.2*	0.266	0.28
TREC7	0.183	0.187*	0.412	0.404
TREC8	0.248	0.249	0.452	0.456

Data	MAP		Precision @10	
	BL	MI	BL	MI
AP-90	0.246	0.264*	0.357	0.381
SJMN	0.188	0.197*	0.252	0.267
TREC7	0.165	0.172	0.354	0.362
TREC8	0.236	0.244*	0.428	0.436

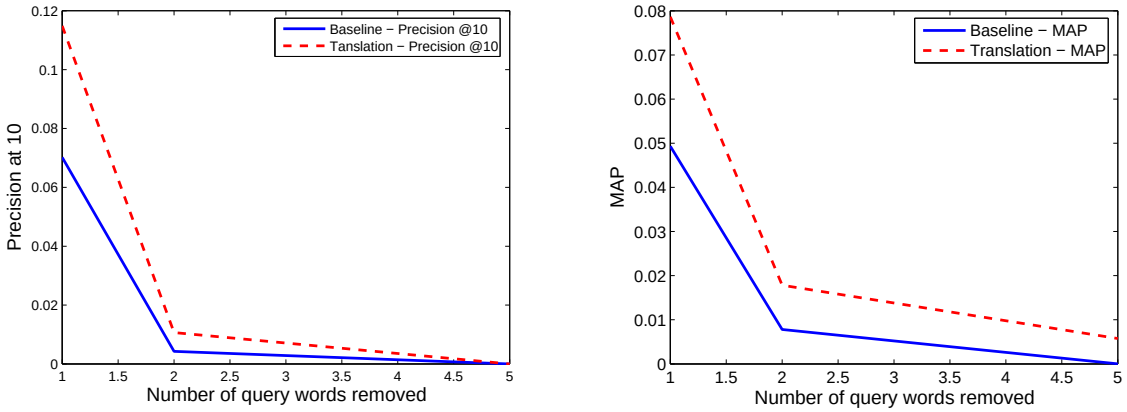

Figure 3.3: Stress Tests on AP90 collection, Precision @10 (left) and MAP (right).

3.5.3 Comparing Translation Language Model with Standard Query Likelihood

We now look into how well a translation model with our mutual information-based estimation method performs as compared with the standard query likelihood method. Table 3.3 shows the results for BL and MI methods according to both measures MAP and Precision @10.

Comparing the columns MI with BL in both tables indeed indicates that the MI outperforms method BL. Significant tests using Wilcoxon signed-rank test [153] show the difference between these two methods for cases marked in the tables are statistically significant. Comparing MI with Dirichlet prior smoothing and MI with JM smoothing shows that MI with Dirichlet prior smoothing has higher MAP than MI with JM smoothing.

Stress Tests: In order to have a better understanding of the translation language model, we applied some stress tests on AP90 data set[3]. This experiment is to help us understand when exactly the translation

[3]The same trends on other data sets have observed, but only the results for AP90 data set is shown.

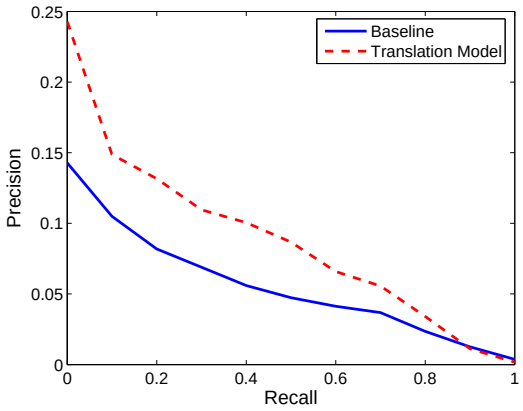

Figure 3.4: Precision-Recall curve when only "one query word" is removed from relevant documents.

language model would be most beneficial. For the stress test, we gradually and randomly remove query words from relevant documents and compare the performance of BL method with MI method. The results of MAP and Precision @10 are shown in Figure 3.3. Figure 3.4 shows the Precision-Recall curve when only "one query word" is removed from relevant documents.

The results indeed indicate that the baseline method (BL) is purely based on exact matching and the performance will drop significantly if the exact matching does not happen. On the other hand, translation language model (MI) is still able to find relevant documents by translating query words to semantically related words in the documents. This indicates that the translation language model works significantly better than the baseline when there is a vocabulary gap between queries and documents.

3.5.4 Effect of Smoothing on Translation Language Model

Understanding the influence of smoothing on translation language model is important and no previous work has looked into this. We have a good understanding of smoothing methods for basic language models [165], but it is not clear how smoothing affects the performance of statistical translation language models. In this section, we look into how statistical translation model behaves with the smoothing parameters.

We vary the smoothing parameters (both JM and Dirichlet prior smoothing) for both BL and MI methods. Figure 3.5 (left and right) shows the variation of the JM smoothing parameter and Dirichlet prior smoothing parameter on AP90, respectively (The results on other data sets are not shown since they are similar.). The result of MI with JM smoothing indicates that the translation model does need a very little smoothing. As shown, the optimal values for translation language model with Dirichlet prior smoothing is 1000 and with JM smoothing is 0.1. As a result, translation language model is less sensitive to the choice of smoothing

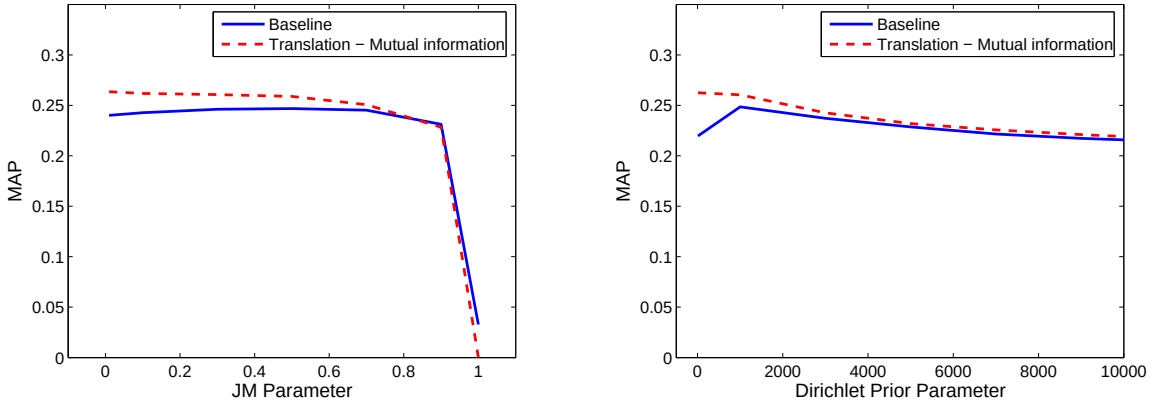

Figure 3.5: JM parameter variation on AP90 (left), Dirichlet prior parameter variation on AP90 (right).

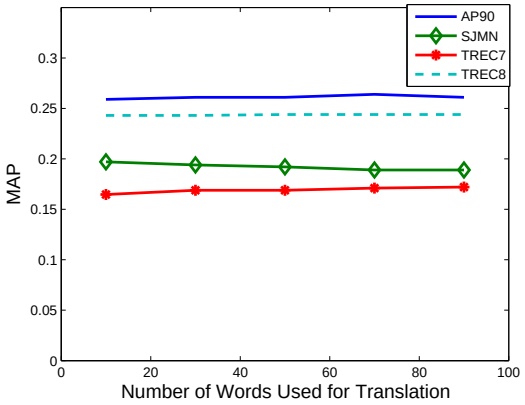

Figure 3.6: Sensitivity of MAP measure to the number of words used for translation.

parameter than the baseline method. And this is intuitively expected, as smoothing is implicitly gained by translating a document word to other *semantically related words*.

Please note that in the translation language model, we have one other parameter to tune, i.e., the number of words used for translation. Figure 3.6 shows the sensitivity of the number of the words according to MAP measure. As shown in the figure, the translation language model is not so sensitive to the number of words used for translation.

3.5.5 Results with Pseudo-Relevance Feedback

Both statistical translation model and pseudo-relevance feedback are to capture word associations, so it would be interesting to see whether they are essentially taking advantage of the same associations or they can be combined to achieve even more improvement.

Table 3.4: Performance of translation language model combined with pseudo-feedback with Dirichlet prior smoothing (top) and JM smoothing (bottom), * and + mean improvements over baseline and fb, respectively, are statistically significant with Wilcoxon signed-rank test. We only show the significance tests for MAP measure.

Data	MAP			Precision @10		
	BL	fb	fb+TM	BL	fb	fb+TM
AP-90	0.248	0.285	0.285*	0.3978	0.404	0.406
SJMN	0.195	0.231	0.232*	0.266	0.295	0.3
TREC7	0.183	0.226	0.226*	0.412	0.38	0.38
TREC8	0.248	0.270	0.278*	0.452	0.456	0.438

Data	MAP			Precision @10		
	BL	fb	fb+TM	BL	fb	fb+TM
AP-90	0.246	0.271	**0.298***+	0.357	0.383	0.411
SJMN	0.188	0.229	**0.234***+	0.252	0.316	0.313
TREC7	0.165	0.209	**0.222***+	0.354	0.38	0.384
TREC8	0.236	0.240	**0.281***+	0.428	0.4	0.452

Table 3.4 shows the pseudo-relevance feedback results for baseline (fb) and when pseudo-relevance feedback is combined with translation language model (fb+TM). For fb+TM method, we first apply pseudo-relevance feedback on initial results (i.e., KL-divergence retrieval model [78]), and then this new query model from pseudo-relevance feedback is used with translation language model to score documents. The feedback parameters are fixed to extract 20 expanded words from the top 10 retrieved documents in the initial run. As shown in table 3.4, fb-TM method indeed outperforms fb method when used with JM smoothing. Statistical significant tests reveal that the difference is indeed statistically significant. However, fb+TM method does not significantly outperform fb method when used with Dirichlet prior smoothing. An interesting observation is that although the performance of pseudo-relevance feedback (fb) method with JM smoothing is lower than pseudo-relevance feedback with Dirichlet prior smoothing, when pseudo-relevance feedback (fb) is combined with translation language model, i.e., fb+TM method, the better performance is gained with JM smoothing. In fact, the performance of fb+TM with JM smoothing is consistently better than the fb+TM with Dirichlet prior smoothing.

Figure 3.7 (left) shows the Precision-Recall curves for BL, fb and fb+TM methods with JM Smoothing on AP90[4]. This figure indeed indicates that the precision of fb+TM method at different recall points is higher than BL and fb methods. This is an interesting conclusion that translation language model brings in co-occurrence word knowledge that once combined with pseudo-relevance feedback, significant improvement is gained.

[4]We do not show other curves due to their similarity.

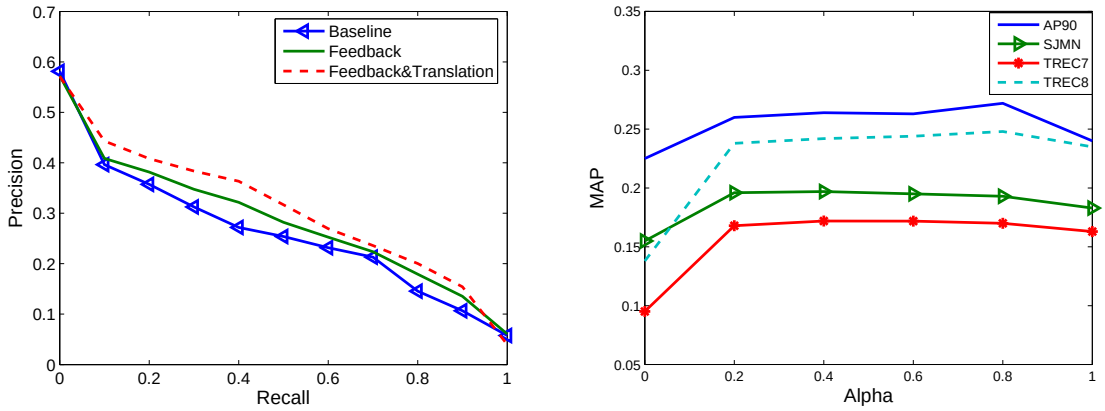

Figure 3.7: Comparison of baseline with translation language model combined with pseudo-relevance feedback and pseudo-relevance feedback alone on AP90 data set with JM smoothing (left) and the sensitivity of MAP measure to α parameter (right).

3.5.6 The Need for Self-Translation Regularization

A potential problem of the estimated translation probabilities is that it is possible that $p(w|u) > p(w|w)$. This may lead to non-optimal retrieval performance because it is possible that a document that matches a query word exactly ($p(w|w)$) gets less score contribution from matching the query word exactly than a document that "matches" a query word through translation ($p(w|u)$). The interpolation formula (with α) can help alleviate this problem; indeed, if $\alpha \geq 0.5$, we can always ensure that this constraint be satisfied. So, it would be interesting to see how α affects the performance. Figure 3.7 (right) shows the sensitivity of MAP measure to α parameter. We indeed observe that when α is very small (close to no interpolation) the performance is poor, suggesting that it is important to regulate the self-translation probabilities to ensure that it is sufficiently large. In Figure 3.7 (right), we can see that when $0.5 \leq \alpha \leq 0.8$ for most data sets, we can gain the optimal value. Note that when $\alpha = 1$, we reach the baseline.

3.6 Chapter Summary

As a principled approach to capturing semantic relation of words in information retrieval, statistical translation models have been shown to outperform simple language models which rely on exact matching of words in the query and documents. In this chapter, we proposed a new simple way to estimate translation probabilities based on mutual information. Our experiment results indicate that the proposed mutual information estimation method is both more efficient and more effective than the existing synthetic query estimation method. The findings are as follows:

1. Translation language model is a principled way to support semantic matching of related words, thus it is

an important contribution in extending the basic query likelihood retrieval model.

2. Translation language model is statistically significant better than the baseline query likelihood especially when there is a vocabulary gap.

3. Normalized mutual information can be used for word-to-word translation effectively and the results in the previous sections indicate that it is more accurate than synthetic queries. Synthetic queries are inefficient for a large collection such as TREC7 or TREC8.

4. The performance of translation language model combined with pseudo-relevance feedback outperforms pseudo-relevance feedback alone; this indicates that translation language model brings in co-occurrence knowledge in addition.

5. Translation language model is less sensitive to the choice of smoothing parameter than the baseline.

6. Translation language model is robust as it improves over all individual queries.

For future, it would be interesting to combine document clustering with translation language model to see if it improves the estimate of translation language models from the viewpoint of smoothing. As it is shown in [87, 150], documents are clustered (either using a cosine similarity [87] or using the Latent Dirichlet Allocation (LDA) [150]) and each document is smoothed with the cluster containing the document and it has been shown that such a smoothing based on clustering improves the performance over the non-clustering smoothing methods [165] as it helps to bring in closely related words which would be really interesting for the estimation in translation language model.

As we discussed in this chapter, the retrieval effectiveness of a translation language model would be mainly determined by the accuracy of the estimated translation probabilities. The question is that how we should know whether the estimated probabilities are optimal. In the next chapter, we perform axiomatic analysis of translation language model to reveal important properties that any estimated translation method should satisfy to have an optimal retrieval behavior .

Chapter 4

Pre-Retrieval: Axiomatic Analysis of Translation Language Model

As shown in the previous chapter, statistical translation language models have been shown to outperform simple document language models which rely on exact matching of words in the query and documents by reducing the vocabulary gap between documents and queries. A main challenge in applying translation language models to ad hoc information retrieval is to estimate a translation model without training data. In the pre-retrieval stage, the retrieval effectiveness of a translation language model would be mainly determined by the accuracy of the estimated translation probabilities, i.e., $p(w|u)$ in the case of standard ad hoc retrieval setting where there is no training data available for learning word-to-word translation probabilities. As shown in the previous chapter, mutual information-based estimation method is more efficient than synthetic query-based approach, however, the translation probabilities estimated using mutual information are based on heuristic normalization of mutual information values with no clearly defined event space associated. This may cause unfair assignment of probability values to different word pairs, and does not provide guidance on how to further improve the estimate.

The main challenging question is that: How do we know whether one estimated translation language model is better than another? Empirical evaluation can only offer limited answer to this question as how well we can generalize an observation on a test set to another is always a concern. Axiomatic analysis [38] has been shown to be useful for diagnosing weakness of a retrieval model and obtaining insights about how to improve a model. In this chapter, we perform axiomatic analysis to gain insights about how to optimize the estimation of a translation language model.

We first prove that in order to behave reasonably for retrieval, the self-translation probability of any word must be a constant. We further define four constraints that a reasonable translation language model should satisfy. We check these constraints on the existing translation language model estimation method based on mutual information and show that it does not satisfy most of the constraints, suggesting that there is room for further improving the estimate. We then propose a new estimation method which is shown to be able to better satisfy the constraints.

4.1 General Translation Language Model Constraints

In this section, we define some general constraints that must be satisfied regardless of any estimation method for translation language model.

Constraint 1: $\forall v$ and $w, p(w|w) = p(v|v)$.

The constraint says that self-translation probability (e.g. translating a document word w to a query word w) for all words should be the same (a constant). This constraint can be justified by the following theorem.

Theorem 1: *In order to have a reasonable retrieval behavior, for all translation language models, we have* $p(w|w) = p(v|v), \forall w, v$.

Proof by Contradiction: Suppose there exist two words w and v such that $p(w|w) \neq p(v|v)$ and consider two documents D_1 and D_2 with identical length and a query with two query terms w and v. Suppose D_1 matches w once, but not v, and D_2 matches v once, but not w. Further assume that w and v have identical probability according to the collection language model, i.e., $p(w|\mathcal{C}) = p(v|\mathcal{C})$. Normally, the two documents would get the same score. But with a translation language model, one might get a higher score than the other "even" if no other words in these documents can be translated into w or v (had there been other words inexactly matching w or v, this difference would be reasonable).

Formally, in the case described above, the score of D_1 would be (assuming there are no other words that can be translated to word w, i.e., $\forall u \in D_1$, if $u \neq w$, $p(w|u) = 0$):

$$\begin{aligned} p(w, v|D_1) &= (\frac{|D_1|}{|D_1| + \mu}[\sum_u p(u|D_1)p(w|u)] + \frac{\mu}{|D_1| + \mu}p(w|\mathcal{C})) * (\frac{\mu}{|D_1| + \mu}p(v|\mathcal{C})) \\ &= (\frac{|D_1|}{|D_1| + \mu}p(w|D_1)p(w|w) + \frac{\mu}{|D_1| + \mu}p(w|\mathcal{C})) * (\frac{\mu}{|D_1| + \mu}p(v|\mathcal{C})) \end{aligned} \tag{4.1}$$

Similarly,

$$p(w, v|D_2) = (\frac{|D_2|}{|D_2| + \mu}p(v|D_2)p(v|v) + \frac{\mu}{\mu + |D_2|}p(v|\mathcal{C})) * (\frac{\mu}{\mu + |D_2|}p(w|\mathcal{C})) \tag{4.2}$$

where $\mathcal{C}$ is the collection and $p(v|\mathcal{C})$ means the smoothed probability of word v based on the collection language model (the same applies to $p(w|\mathcal{C})$).

Since $p(w|\mathcal{C}) = p(v|\mathcal{C})$, $p(w|D_1) = p(v|D_2)$ and $|D_1| = |D_2|$, the score ratio would be

$$\frac{p(w, v|D_1)}{p(w, v|D_2)} = \frac{p(w|D_1)p(w|w) + p(w|\mathcal{C})}{p(w|D_1)p(v|v) + p(w|\mathcal{C})} \tag{4.3}$$

That is, if w has a higher self-translation, D_1's score would be higher which is not reasonable. Thus, in order to avoid such unreasonable behavior, we must have: $\forall w$ and v, $p(w|w) = p(v|v)$, which provides the proof for theorem.

This is an interesting new analytical result, and without special attention, in general, there is no guarantee that the self-translation probability would be the same for all the words. As we will show in the experiments, if we heuristically modify an existing translation model to make it satisfy this constant constraint, we will be able to improve the retrieval accuracy.

Furthermore, the self-translation probability should be larger than translation probabilities involving different words which leads to the following two constraints:

Constraint 2: $\forall w, u$, if $w \neq u$, then $p(w|w) >= p(w|u)$.
This constraint states that self-translation probability (i.e., translating a document word w to a query word w) should be larger than translating any other words to this word (i.e., translating a document word u to a query word w). From retrieval perspective, suppose that a query term is w, and further assume that a document D_1 contains word w, and no other words in D_1 can be translated to w, so the score of D_1 would be $\frac{|D_1|}{|D_1|+\mu}p(w|w)p(w|D_1)+\frac{\mu}{|D_1|+\mu}p(w|\mathcal{C})$. In another document D_2, assume that it does not have any occurrence of word w but has the same number of occurrence of u as w in D_1, i.e., $c(w, D_1) = c(u, D_2)$ and the two documents have the same length. Similarly, the score of D_2 would be $\frac{|D_2|}{|D_2|+\mu}p(w|u)p(u|D_2) + \frac{\mu}{|D_2|+\mu}p(u|\mathcal{C})$. It is clear that the first case is the exact match while the second one is the inexact match. Since $|D_1| = |D_2|$, $p(w|D_1) = p(u|D_2)$ and $p(w|\mathcal{C}) = p(u|\mathcal{C})$, if this constraint is not satisfied, the score of D_1 would be smaller than the score of D_2 which is undesirable. So, this constraint must be satisfied.

Constraint 3: $\forall w, u$, if $w \neq u$, then $p(w|w) >= p(u|w)$.
This constraint intuitively means that a word is more likely to be translated to itself (i.e., $p(w|w)$ which means that a document word w is translated to a query word w) than translating into any other words (i.e., $p(u|w)$ which means a document word w is translated to a query word u) which makes sense.

4.2 Additional Translation Language Model Constraints

Since co-occurrences are the primary information used for estimating translation models without training data, in this section, we further propose two additional constraints based on word co-occurrences that any reasonable estimation method for translation language model should satisfy. Note that $c(x, y)$ means the number of co-occurrences of word x with word y in a context. We counted the number of co-occurrences in the context of documents but any other contexts such as sentences, etc can also be used.

Constraint 4: if $c(w, u) > c(w, v)$ and $\sum_{w'} c(w', u) = \sum_{w'} c(w', v)$ then $p(w|u) > p(w|v)$.

Constraint 5: if $c(w, u) = c(w, v)$ and $\sum_{w'} c(w', u) < \sum_{w'} c(w', v)$ then $p(w|u) > p(w|v)$.

The fourth constraint states that if word u occurs more times than v in the context of w and both words u and v co-occur with every other words similarly, the probability of translating word u to word w should be higher than that of translating v to word w. From retrieval perspective, if a document mentions word u, $p(w|u)$ should capture the probability that the document can be regarded as matching a query term w. In other words, the presence of u in a document more likely implies that the document matches w in the query, than the presence of v in the document does. Consider the following example. Suppose w ="Europe", u ="France" and v = "Chicago". We can expect "Europe" to co-occur more with "France" than with "Chicago", thus $c(w, u) > c(w, v)$. This constraint basically requires p(Europe|France) to be larger than p(Europe|Chicago).

The fifth constraint states that if both words u and v equally co-occur with word w but v co-occurs with many other words than word u, the probability of translating word u to w is higher. Intuitively, this means that word v is a more general word than u since it co-occurs with so many other words, so the confidence of translating to word w is less because it could potentially be translating into many other words.

4.3 Analysis of Mutual Information-Based Translation Language Model

In this section, we analyze the existing state of the art translation language model which is based on mutual information, i.e., Equation 3.13 to see if it satisfies the proposed constraints.

Constraint 1: In general, this method does not ensure the self-translation probabilities are constant across all the words. The need for sufficient self-translation probability was addressed in Section 3.4.2 and

a regularized parameter α is introduced to control the effect of self-translation. We call this method "MI" in our experimental results.

Although, this heuristic modification helps satisfy the second constraint, it still does not ensure the constant self-translation probabilities. We will define a general heuristic way to make the self-translation probabilities constant later in the dissertation.

Constraint 2: The analysis of the second constraint is complicated. However, the following examples (taken from DOE data set) show that mutual information does not satisfy it. Note that the words are stemmed.

p(microeconom|microeconom) = 0.3362 and p(microeconom | fluctuact) = 0.4421.

p(sneez | sneez) = 0.1107 and p(sneez | rhinorrhea) = 0.1371.

However, with the α regularization, this constraint is satisfied.

Constraint 3: We need to show that $\frac{I(w;w)}{\sum_{w'} I(w';w)} > \frac{I(w;u)}{\sum_{w'} I(w';w)}$. Since $I(w;w) > I(w;u)$, it is easy to see that this constraint is always satisfied.

Constraint 4: The analysis of this constraint is complicated. However, the following examples (taken from DOE data set) show that mutual information does not satisfy this constraint. Note that the words are stemmed.

1: c(deliber, ensur)=24 > c(deliber, recogn) = 22 and ($\sum_{w'}$ c(w', ensur) = $\sum_{w'}$ c(w', recogn) = 4105) but p(deliber|recogn) = 0.0018 > p(deliber|ensur) = 0.0017.

2: c(hypothermia, hypotherm) = 4 > c(hypothermia, mepyramin)= 3 and ($\sum_{w'}$ c(w', hypotherm) = $\sum_{w'}$ c(w', mepyramin) = 21) but p(hypothermia|mepyramin) = 0.073 > p(hypothermia|hypotherm) = 0.0669.

Constraint 5: This constraint is not satisfied since the followings are a few undesirable examples taken from DOE collection (words are stemmed):

1: c(theolog, benefit) = c(theolog, held) and ($\sum_{w'}$ c(w', benefit) = 5211) < ($\sum_{w'}$ c(w', held) = 7496) but p(theolog| benefit) = 0.0001 < p(theolog| held) = 0.00014.

2: c(polytomograph,neck) = c(polytomograph,thirti) and ($\sum_{w'}$ c(w', neck) = 1652) < ($\sum_{w'}$ c(w', thirti) = 1710) but p(polytomograph |neck) = 0.00046 < p(polytomograph|thirti)= 0.00066.

The fact that the mutual information-based method does not satisfy at least four of the proposed five

constraints suggests that there may be room for improving the estimation of translation language model. In the next section, we present a new estimation method based on conditional context analysis.

4.4 Estimation of Translation Probabilities based on Conditional Context Analysis

In this section, we propose a new method that can better satisfy the constraints. As mentioned before $p(w|u)$ would give us the probability that a document containing u can be regarded as matching a query term w. In other words, this is the probability that matching a document word u implies matching a query word w, or if a document is about u, it is also about w. To illustrate this, consider two semantically related words "Europe" and "France". Intuitively, if a document mentions "France", we can confidently say that it is also about "Europe", thus we should expect $p(Europe|France)$ to be relatively high. In contrast, if a document mentions "Europe", it does not necessarily imply that it is about "France" (since it can be about any other European country), thus we should expect $p(France|Europe)$ to be somewhat low. The reason why we expect $p(Europe|France)$ to be much larger than $p(France|Europe)$ is because "Europe" is a broader concept than "France". Indeed, if we look at all the documents where "France" occurs, we may expect to see relatively high frequency of "Europe"; on the other hand, "France" does not necessarily occur frequently in the documents that contain "Europe". This analysis suggests that we can use word context analysis to capture the desired asymmetry. More specifically, we can use the frequency of seeing word w in the context of word u to estimate $p(w|u)$ so that if we see w often in the context of u, $p(w|u)$ would be high, whereas if w does not occur in the context of u, $p(w|u)$ would be nearly zero. Context can be any meaningful text units such as a sentence, a paragraph, or a document. In our experiments, we assume that a document is a context unit. This forms the basic idea of our new estimation method. Below, we present the method more formally:

Let $c(u) = n$ be the number of all context units containing word u in our collection; we estimate $p(w|u)$ based on the conditional probability of seeing w in the context of u, i.e., $p(w|u) = \frac{c(w,u)}{n}$ where $c(w,u)$ is the number of times that these two words u and w co-occur with each other in a context unit. In order to account for unseen words in the context of u, we do *Additive smoothing* [165]. As a result, $p(w|u)$ is estimated as follows:

$$p(w|u) = \frac{c(w,u)+1}{\sum_{w'} c(w',u) + |V|} \tag{4.4}$$

Where $\sum_{w'} c(w',u)$ is the co-occurrences of word u with other words in the vocabulary (i.e., w') in the

context of u and $|V|$ is the size of the vocabulary.

We now analyze the five constraints for this new estimation method:

Constraint 1: As in the case of mutual information-based approach, this method does not ensure the self-translation probabilities are constant across all the words.

Constraint 2: For satisfying the second constraint, we should show that $\frac{c(w,w)+1}{\sum_{w'} c(w',w)+|V|} > \frac{c(w,u)+1}{\sum_{w'} c(w',u)+|V|}$. After some algebraic transformation, it is equivalent to show that:

$$|V| > \frac{[(\sum_{w'} c(w',w))(1+c(w,u))] - [(\sum_{w'} c(w',u))(1+c(w,w))]}{c(w,w)-c(w,u)} \tag{4.5}$$

This suggests that as long as we set $|V|$ to a sufficiently large value, this constraint would be satisfied. However, can we obtain a lower bound for $|V|$? To investigate this, we may consider a special case where the denominator is minimized, i.e., $c(w,w) = c(w,u) + 1$ and $\sum_{w'} c(w',w) = c(w)|D|$ where $|D|$ is the average document length. This leads to: $|V| > [c(w)|D| - c(u)|D|)]c(w,w) - c(u)|D|$. This gives us a way to empirically set $|V|$ to a sufficiently large value.

Constraint 3: We need to show that $\frac{c(w,w)+1}{\sum_{w'} c(w',w)+|V|} > \frac{c(w,u)+1}{\sum_{w'} c(w',w)+|V|}$, since $c(w,w) > c(w,u)$, the constraint is always satisfied.

Constraint 4: We need to show that $\frac{c(w,u)+1}{\sum_{w'} c(w',u)+|V|} > \frac{c(w,v)+1}{\sum_{w'} c(w',v)+|V|}$, Since the denominators are equal (i.e., $\sum_{w'} c(w',u) = \sum_{w'} c(w',v)$) and $c(w,u) > c(w,v)$, this constraint is satisfied.

Constraint 5: We need to show that $\frac{c(w,u)+1}{\sum_{w'} c(w',u)+|V|} > \frac{c(w,v)+1}{\sum_{w'} c(w',v)+|V|}$, Since $c(w,u) = c(w,v)$, and $\sum_{w'} c(w',u) < \sum_{w'} c(w',v)$, it is easy to see that this constraint is satisfied.

We call this method "Cond" in our experimental results. We see that the new method better satisfies the constraints than mutual information-based approach. As we will show later, the new method indeed outperforms mutual information-based approach.

4.5 Heuristic Adjustment of Self-Translation Probability

As pointed out earlier, none of the methods "MI" and "Cond" satisfy the first constraint. Here, we present a general heuristic way to make the self-translation probability constant, which we will show later improves over

the self-translation proposed in Section 3.4.2 (with α parameter). Given any estimated translation model $p(u|v)$, we can define an adjusted translation language model $p'(u|v)$ to ensure constant self-translation probability as follows:

$$p'(u|u) = s(s >= 0.5) \tag{4.6}$$

Replacing $p'(u|u) = s$ in Equation 3.15 gives us:

$$\alpha + (1 - \alpha) * p(u|u) = s \Rightarrow \alpha = (s - p(u|u))/(1 - p(u|u)). \tag{4.7}$$

Then plugging in α in Equation 3.15 gives us:

$$p'(w|u) = (1 - s) * p(w|u)/(\sum_{v \neq u} p(v|u)). \tag{4.8}$$

We then plug in Equations 4.6 and 4.8 in Equation 3.7 to replace $p(w|u)$.

We can apply this strategy to both methods "MI" and "Cond" leading to two additional methods, which we denote as "CMI" and "CCond" in our experiments.

4.6 Experiments

The experiments in this section use three main document collections: (1) ad hoc data in TREC7 with TREC topics 351-400 and 528,155 articles (2) WSJ news articles with TREC topics 51-100 and (3) technical reports in DOE abstracts with TREC topics 51-100.

In the experiments, we only use title of the queries because short keyword query is the most frequently used query type by web users and semantic term matching is necessary for such short queries. For all data sets, preprocessing of documents and queries is minimum and involves only stemming with Porter stemmer [104] and stop word removal. All experiments are done using the Lemur toolkit. The performance is measured using two standard measures: MAP and Precision@10. MAP serves as our main measure but we show the results for P@10 for the sake of completeness. We set Dirichlet prior smoothing for baseline methods to 1000 for all data sets. The methods used for the experiments in the following sections are:

BL: BaseLine, i.e., basic query likelihood method.

MI: Mutual information, i.e., translation language model with mutual information-based approach with parameter α for self-translation probabilities explained in section 3.4.2.

Table 4.1: Performance of translation language model on different datasets with conditional-based approach (top): cross validation and (bottom): upper bound, * and + mean improvements over baseline BL and MI are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Data	**MAP**			**Precision @10**		
	BL	MI	Cond	BL	MI	Cond
TREC7	0.1852	0.1854	**0.1864***+	0.4180	0.42	0.418
WSJ	0.2600	0.2658	**0.275***+	0.424	0.44	0.448
DOE	0.1740	0.1750	**0.1758***	0.1913	0.1956	0.2043

Data	**MAP**			**Precision @10**		
	BL	MI	Cond	BL	MI	Cond
TREC7	0.1852	0.1885	**0.1887***	0.4180	0.42	0.446
WSJ	0.2600	0.2708	**0.2778***+	0.424	0.44	0.448
DOE	0.1740	0.1813	**0.1868***+	0.1913	0.1956	0.2086

Cond: Conditional context, i.e., translation language model with conditional-based approach with parameter α for self-translation probabilities.

CMI: Constant mutual information, i.e., translation language model with constant parameter s for self-translation probabilities explained in section 4.5 and mutual information-based approach for word-to-word translation probabilities.

CCond: Constant conditional context, i.e., translation language model with constant parameter s for self-translation probabilities explained in section 4.5 and conditional-based approach for word-to-word translation probabilities.

For the translation language model, we have two parameters 1) the number of words used for translation 2) s: the amount of self-translation probabilities in Equations 4.6 and 4.8. In order to have a fair comparison of the methods, we do leave-one-out cross validation to learn the two parameters for our method.

4.6.1 Comparing Conditional-Based Approach with Baselines

Our first hypothesis is that the proposed conditional-based approach for word-to-word translation probabilities with parameter α for self-translation probabilities (method Cond) should outperform the BL method and state of the art method MI which is the mutual information-based approach for word-to-word translation probabilities with parameter α for self-translation probabilities. Table 4.1(top) shows the leave-one-out cross validation results for these methods and table 4.1(bottom) shows the upper bound results.

Comparing the columns Cond with BL in both tables indicates that the Cond method outperforms the BL method according to both MAP and P@10. Significant tests using Wilcoxon signed-rank test[153] show that the difference between these two methods are statistically significant. Since translation language model

bridges the vocabulary gap and it is not only based on exact matching of query words, these results are expected. In addition, the results confirm our hypothesis that conditional-based approach is able to capture the word relatedness.

Comparing the columns MI with Cond in both tables indicates that the Cond outperforms the MI method. Significant tests using Wilcoxon signed-rank test show that the difference between these two methods are statistically significant for cases marked in the table. These results are intuitively expected which indicates that in order to have a reasonable retrieval behavior, estimation methods should satisfy all the constraints and we see that method Cond satisfies more constraints (it does not satisfy only the first constraint) than method MI.

Table 4.2 (left) and (right) show some sample words that can be translated to word "launch" (a query word in TREC 54) based both on conditional-based approach and mutual information-based approach, respectively. As it is shown since the conditional-based approach considers the context of the words into account, the other words that "launch" can be translated to are all in the same context whereas for the mutual information-based approach, the words are not necessarily in the same context which further indicates that the proposed estimation method based on conditional probabilities is better than mutual information.

Table 4.2: Sample word translation probabilities using conditional-based approach (left) and mutual information-based approach (right). Note that words are stemmed. "launch" is a query word in TREC query 54.

w=launch	
u	$p(w\|u)$
launch	0.00341
nasa	0.00087
rocket	0.00082
tender	0.00078
satellit	0.00078
shuttl	0.00073
orbit	0.00072
eurobond	0.00069
missil	0.00068
space	0.00061

w=launch	
u	$p(w\|u)$
launch	0.59830
rocket	0.02382
nasa	0.0222
shuttl	0.01675
satellit	0.01546
aim	0.01351
target	0.01117
space	0.01085
rival	0.00926
hostil	0.00839

4.6.2 Comparing Methods with Constant Self-Translation Probability

In this section, we look into the question whether the constant self-translation probability is effective for both methods MI and Cond. The results are shown in Table 4.3 for both leave-one-out cross validation results (top) and upper bound results (bottom). Comparing columns of MI and CMI clearly shows that the

Table 4.3: Performance of translation language model on different datasets with conditional-based approach and mutual information-based approach (top): cross validation and (bottom): upper bound, *, + and & mean improvements over Cond, MI and CMI are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Data	**MAP**				**Precision @10**			
	MI	CMI	Cond	CCond	MI	CMI	Cond	CCond
TREC7	0.1854	0.1872^{+}	0.1864	**0.1920***&	0.42	0.408	0.41	0.418
WSJ	0.2658	0.267^{+}	0.275	**0.278***&	0.44	0.442	0.448	0.448
DOE	0.1750	0.1774^{+}	0.1758	**0.1844***&	0.1956	0.2	0.2043	0.2

Data	**MAP**				**Precision @10**			
	MI	CMI	Cond	CCond	MI	CMI	Cond	CCond
TREC7	0.1885	0.1905^{+}	0.1887	**0.1965***&	0.42	0.41	0.41	0.418
WSJ	0.2708	0.2717^{+}	0.2778	**0.2800***&	0.44	0.448	0.448	0.45
DOE	0.1813	0.1841^{+}	0.1868	**0.1953***&	0.1956	0.2043	0.2086	0.2086

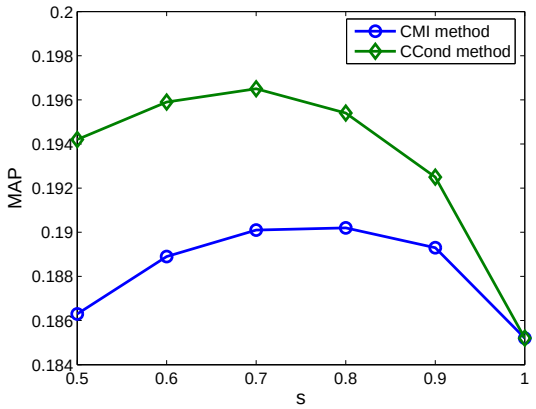

Figure 4.1: Comparison of mutual information-based approach and conditional-based approach according to MAP measure on TREC 7 data set.

constant self-translation probabilities are effective. The statistical significant tests using Wilcoxon signed-rank test shows that the results are indeed significant. In addition, comparing the columns of Cond and CCond shows that CCond performs better than Cond stating that the constant self-translation probability is indeed helpful.

In addition, comparing our proposed method CCond with both MI and CMI clearly indicates that the conditional-based approach for translation probabilities with constant self-translation probability helps improve the performance. Figure 4.1 shows the sensitivity of conditional-based approach and mutual information-based approach to parameter s according to MAP measure on TREC 7 data set[1]. The difference indeed makes clear that conditional-based approach works better than mutual information-based approach.

[1]Similar trends can be seen for other data sets.

4.6.3 Results with Pseudo-Relevance Feedback

Pseudo-relevance feedback and translation language models both capture the word associations. The both estimation methods, i.e., mutual information-based approach and conditional-based approach are considered global methods as they consider the whole collection to estimate the translation probabilities. However, pseudo-relevance feedback is considered a local method, as it only considers the words in the top-ranked documents to expand the query. As shown in Section 3.5.5, combining translation language model with feedback helps improve the performance more. So, our hypothesis is that the proposed method CCond when combined with feedback can still outperform method fb+MI. Table 4.4 shows the pseudo-relevance feedback results for baseline, when pseudo-relevance is combined with mutual information-based translation language model (fb+MI) and fb+CCond (constant conditional-based approach combined with feedback). We again show the results based both on leave-one-out cross validation (top) and upper-bound (bottom) results. For both methods fb+MI and fb+CCond, we first apply pseudo-relevance feedback on initial results (i.e., KL-divergence retrieval model with Dirichlet prior smoothing [165]), and then this new query model from pseudo-relevance feedback is used with both mutual information-based translation language model and conditional-based approach. The feedback parameters are fixed to extract 10 top words from top 20 ranked documents. As it is shown, the method fb+CCond can outperform both fb and fb+MI which further confirms that the proposed estimation method is better than mutual information-based approach. In addition, the differences between the results are statistically significant for the cases marked in the tables.

Table 4.4: Performance of translation language model with conditional-based approach combined with pseudo-relevance feedback on different datasets (top): cross validation and (bottom): upper bound, * and + means improvements over fb and fb+MI are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Data	MAP			Precision @10		
	fb	fb+MI	fb+CCond	fb	fb+MI	fb+CCond
TREC7	0.2203	0.2210	**0.2236***+	0.394	0.396	0.4
WSJ	0.2947	0.295	**0.2975***+	0.47	0.468	0.464
DOE	0.1696	**0.1868**	0.186	0.1956	0.2	0.2

Data	MAP			Precision @10		
	fb	fb+MI	fb+CCond	fb	fb+MI	fb+CCond
TREC7	0.2208	0.2232	**0.2242***	0.394	0.396	0.4
WSJ	0.2947	0.2974	**0.2988***+	0.472	0.468	0.466
DOE	0.1889	0.1921	**0.1927***	0.2043	0.1956	0.213

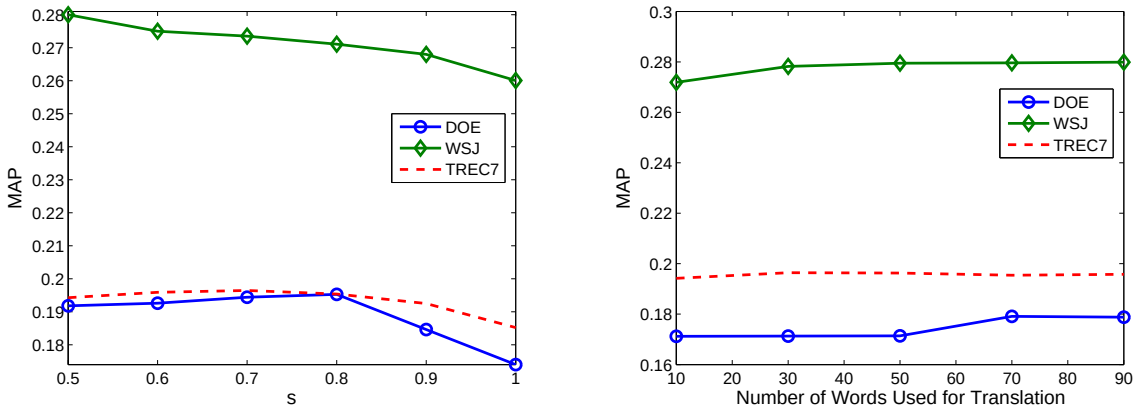

Figure 4.2: Sensitivity of MAP measure to parameter s (left) and sensitivity of MAP measure to the number of words used for translation (right).

4.6.4 Parameter Sensitivity Study

There are two parameters associated with the translation language model which are the number of words and s which controls the amount of self-translation probabilities. In this section, we study how sensitive the MAP measure is to these two parameters.

Figure 4.2 (left) shows the sensitivity of MAP measure to parameter s in conditional-based translation language model (CCond) for all data sets. The figure shows that the parameter s should be set to $0.5 <= s <= 0.8$ to achieve the optimal performance. Please note that when $s = 1$, the baseline (BL) results is gained.

Figure 4.2 (right) shows the sensitivity of MAP measure to the number of words used for translation in conditional-based translation language model (CCond) for all data sets. The figure shows that the method is not so sensitive to the number of words used for translation.

4.7 Chapter Summary

Estimating word-to-word translation probabilities for statistical translation language model is challenging in monolingual information retrieval. In this chapter, we performed axiomatic analysis for translation language model and introduced various constraints that a reasonable translation language model should satisfy. The existing estimation method based on mutual information did not satisfy all of them. We then proposed a new estimation method based on the context of the words that satisfies all the constraints except for the first one. We then defined a heuristic way to make the self-translation probability a constant so that both methods can satisfy the first constraint. Experimental results show that our proposed new estimation is

better than mutual information-based estimation, suggesting that the proposed constraints are probably necessary in order to achieve good performance. Also, for both mutual information-based approach and the new estimation method, when we ensure self-translation probabilities to be constant, it would also improve performance, empirically supporting the proposed constraint.

For future work, it would be interesting to develop more constraints on translation models that can help improve effectiveness of a translation model for retrieval. In addition, setting up a unified optimization framework for estimating translation probabilities where constraints can directly be incorporated into the estimation method is another future direction.

So far, in the last two chapters, we proposed novel methods that help improve the accuracy of retrieval models in the pre-retrieval stage. In the next chapter, we describe an application of the statistical translation language model for Twitter search.

Chapter 5

An Application of Statistical Translation Language Model - Twitter Search

With the prevalence of social media applications, an increasing number of internet users are actively publishing text information online. This influx provides a wealth of both text and social graph information on those users. In weblog posts one can find a wide range of topics being discussed, as well as an evolving and implicit friendship network among the bloggers themselves. In such social networks, we are dealing with collections of text content where they exhibit a number of unique properties that necessitate the development of novel ranking techniques for social media search. While there have been different ranking methods for Web search, there is little study of ranking in social media. In the past, social network analysis solely focused on the topology structure of a network addressing questions such as "what the diameter of a network is [83]", "how a network evolves [83]", and "how information diffuses on a network [84]". However, these techniques usually do not leverage the rich text information.

Ranking in social media poses different challenges than Web search ranking, one of which is that microblog messages are really short. For example, Twitter [143] messages are limited to 140 characters of contents. This limit causes users to write in an abbreviated form [49] which increases the likelihood of having the vocabulary gap (i.e., vocabulary mismatch problem). The vocabulary mismatch/gap problem means that the user query uses one word and the relevant documents use another word and we need to match the query with documents.

In this chapter, we study whether and how we can use statistical translation language model to solve the ranking in Twitter. Twitter has gained huge popularity since the first day that was launched which has drawn increasing interest from research and industry communities. As mentioned in the previous chapters, statistical translation language model has been proposed for ad hoc information retrieval to reduce the vocabulary gap between documents and queries. In this chapter, we first study whether a direct application of standard translation language model would improve the performance for Twitter search and find that it improves the performance. Further analyses reveal that statistical translation language model can have two benefits; it helps to bridge the vocabulary gap and in case when there is no vocabulary gap it still helps by improving the estimate of term frequency (TF) especially for short tweets where the words appear only once

and there is no discrimination between the words. We then propose two ways to improve the performance of statistical translation language model further. The first method is to leverage the hashtag information to bridge the vocabulary gap between remotely related words. The second way is to have a per-term self-translation parameter. We evaluate the proposed methods by comparing them with the best-performing estimation method as described in the previous chapter, i.e., constant conditional context estimation method on Twitter data set (called CCond). Experimental results show that the proposed solutions are effective in alleviating the vocabulary gap problem, thus improving the accuracy of search results for Twitter search.

5.1 DataSet

For our data set, we use TREC microblog track 2011 data set. The microblog track was unique because it required participants to crawl/download the data set on their own. This corpus consists of microblog posts made available by Twitter. TREC organizers distributed data in a two-stage process. First, organizers enumerated a set of approximately 16M unique tweet ID numbers. Participants downloaded these ID's, then downloaded the tweets themselves using a software [135] provided by the track organizers. The TREC-supplied software supported two types of download. We used the version that directly scrapes tweets from Twitter's web-facing HTML. We used this software to download those tweets among the enumerated ID's that were available via HTTP on May 25-26, 2011. In total our data set has 15,653,612 documents with 5,243,118 unique users. Tweets are inherently multi-lingual, but we only consider English tweets in this dissertation and we leave searching in multi-lingual language for future work. Therefore, accurate language identification is important. We apply the following heuristics algorithm to discard non-English tweets from our data set.

Given a set of tweets T, we want to find E $\sqsubseteq$ T such that E consists of only English tweets. We start by defining a mapping R(w) that maps each entry in our set of known English words to a score value of 1.0, and all other strings to 0.0. We then add more non-zero valued mappings to R during multiple passes (we used 2 passes) over T. In each pass, we do the following to each tweet: break up each tweet into distinct words, strip out hashtags, usernames, URLs, special characters and simple repeating patterns. We then look at the average score, v, of the tweet by averaging together the R(word) values of its words. If this aggregate score v is above a particular threshold value (we used 0.7), all the words w_i found in the tweet are given a new score value $R(w_i) \leftarrow \max(R(w_i), v)$, effectively upgrading each word's confidence rating for English usage. On the final pass, we designate all tweets with an aggregate score above a threshold (0.7 again) as

Table 5.1: Comparing the results of translation language model, i.e., CCond with BL.

Method	MAP	P@10
BL	0.21	0.3244
CCond	0.2207	0.3448

containing primarily English content.

This results to contain only English tweets which are 8,349,758 tweets with 3,375,847 unique users. There are about 1,906,548 users that have written only one tweet in total.

A set of 50 test topics was released for evaluation purposes. Each topic consists of a keyword query and a temporal reference point, which acts as a query timestamp. NIST employed a pooling strategy to obtain ground truth for evaluation purposes. There were no relevant items for topic 50 and it was therefore eliminated from the topic pool.

Preprocessing of tweet documents and queries is minimum and involves only stemming with Porter stemmer and stop word removal. All experiments are done using the Lemur toolkit. The performance is measured using two standard measures: MAP and Precision @10. The optimal value for Dirichlet prior smoothing parameter for baseline method is 1000.

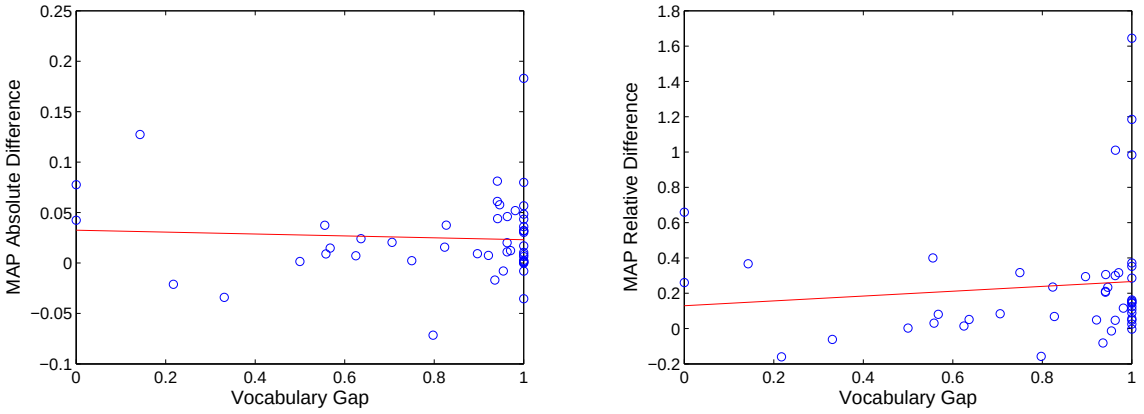

Figure 5.1: Vocabulary Gap measure against absolute difference of the translation language model (CCond) and BL (left) and Vocabulary Gap measure against the relative difference between translation language model (CCond) and BL (right).

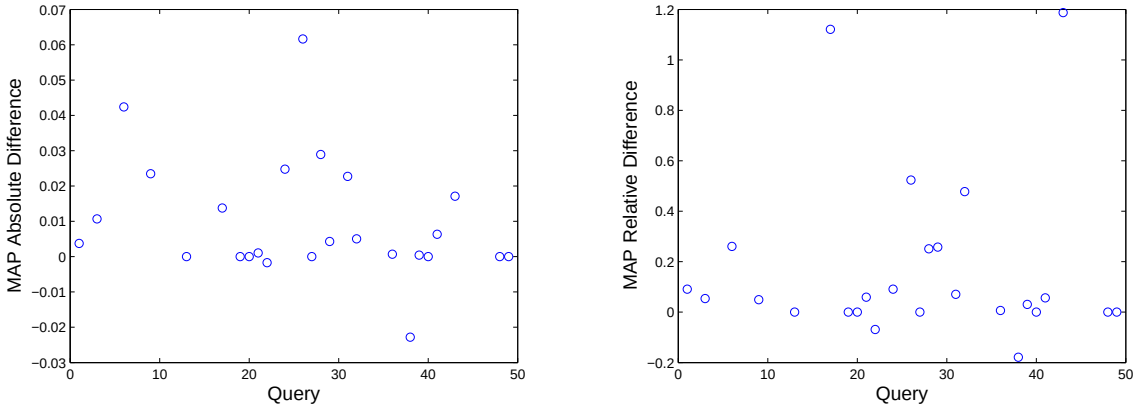

Figure 5.2: Absolute difference of the translation language model (CCond) and BL (left) and relative difference between translation Language model (CCond) and BL (right).

5.2 Study of Standard Translation Model for Twitter Search

In this section, we study the effectiveness of direct application of statistical translation language model for Twitter search. And specifically we describe our vocabulary gap hypothesis:

Since there is a vocabulary gap in tweets, statistical translation language model can be used to reduce the vocabulary gap between documents and queries.

In order to test the vocabulary gap hypothesis, we compare the results of statistical translation language model with the BL approach, i.e., KL-divergence with Dirichlet prior smoothing [165]. The results are shown in Table 5.1. Comparing the results of the statistical translation language model (CCond) with BL method shows that there is an improvement over the BL method. However, given that the average length of the tweets is short, the improvement is not as much as we expected. The question now is whether this kind of model is really effective for bridging the vocabulary gap? to answer this question, we look into the statistical translation language model to see if it really fills in the vocabulary gap for this Twitter data set.

In order to examine the vocabulary gap problem in tweets quantitatively, we propose a new measure, i.e., *Vocabulary Gap* measure and plot the Vocabulary Gap measure against MAP where Vocabulary Gap measure is defined as follows:

Vocabulary Gap measure = 1 - the percentage of the relevant documents that contain all the query words.

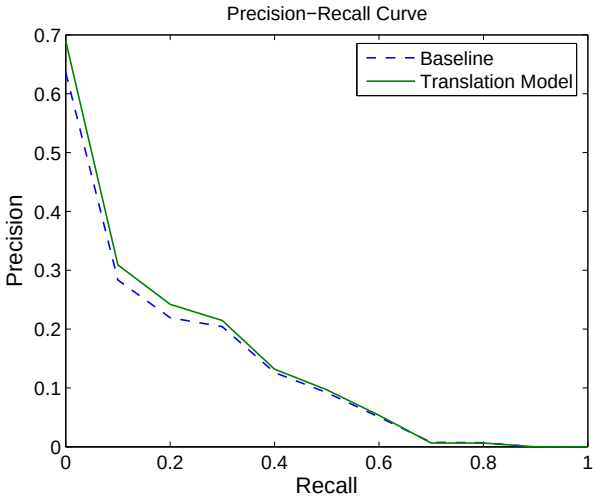

Figure 5.3: Precision-Recall curves comparing statistical translation language model (CCond) with BL method

When Vocabulary Gap = 1, it means that no documents contain all the words and when Vocabulary Gap = 0, it means that all the documents contain all the words which make sense.

We then plot the Vocabulary Gap measure against absolute difference (MAP(CCond)-MAP(BL)) and Vocabulary Gap measure against the relative difference:

$$\text{Relative MAP difference} = \frac{MAP(CCond) - MAP(BL)}{MAP(BL)}$$

The results are shown in Figures 5.1 (left) and (right), respectively. The results of both of these figures indeed indicate that when there is a vocabulary gap (i.e., 1), statistical translation language model is indeed helpful which confirms our hypothesis.

The interesting observation is that even if all (or many) query words are matched, we still observe an improvement of the statistical translation language model. We hypothesize that the benefit comes from improving the estimation of the term frequency heuristics. Indeed translation language model is a way of smoothing and smoothing is important for short tweets, but even though the word occurs in a tweet, the counts of the word is not informative. So, we look into the case to see even if there is no vocabulary gap (i.e., all query terms are matched), statistical translation language model is still helpful because it helps term frequency heuristics.

So for this experiment we look at *only* documents that match all query terms, and see if translation language model can re-rank these documents more accurately than the BL method. We show the results for each query in Figures 5.2 (left) based on absolute difference and (right) based on relative difference. The results confirm that even if there is no vocabulary gap problem, statistical translation language model is still

helpful because it helps to improve the term frequency.

Figure 5.3 shows the Precision-Recall curves comparing the BL method with translation language model for this experiment. The Precision-Recall curves indeed indicate that the statistical translation language model is still helpful even if there is no vocabulary gap.

Table 5.6 (top) shows the ranked lists of tweets for query "NSA" according to the BL method and Table 5.6 (bottom) shows ranked tweets for translation language model. The italic tweet words in translation language model (i.e., Table 5.6 (bottom)) are those words that support the query word "NSA" through the translation language model and it shows that for the top 10 ranked tweets, the BL method can only retrieve one relevant document vs. the translation language model can return three relevant documents. In addition, the BL method generally favors short documents vs. there is not such a limit for statistical translation language model due to getting support from other semantically related words in the tweets and as a result improving the term frequency of the query words.

While in the past the benefit of semantic smoothing achieved by a translation model is mainly to bring in unseen words related to a document, and its benefit for adjusting TF values for seen words in a document may not be significant (since documents are generally long and the observed counts of seen words may be discriminative enough), in Twitter search, the observed counts are not discriminative, and thus benefit of adjusting the TF estimation becomes more significant.

In summary, the study of standard translation language model has two benefits for Twitter search and in the next section, we study if we can further improve the translation language model.

5.3 Further Improving Statistical Translation Language Model

In this section, we investigate to see if we can further improve the statistical translation language model for Twitter search.

5.3.1 Leveraging Hashtag Information

Many tweets are marked with # called hashtags. Hashtags are used to mark the topics in a tweet or the intended audience [34]. For example, #CIKM2012 marks all tweets related to the 2012 CIKM conference. In this section, we show that how hashtags can be leveraged to further improve the statistical translation model by alleviating the data sparseness problem.

Table 5.2 (left) shows the number of hashtags for the number of tweets' category. For example, it

Table 5.2: Hashtag distributions.

Number of Tweets	Number of Hashtags
1−5000	316,703
5000−10000	5
10000−15000	1
15000−20000	3
20000−25000	1

Number of Tweets	Number of Hashtags
1	233,730
2−10	68,760
11−100	12,663
101−5000	1550

states that there are about 316,703 unique hashtags that have between 1 to 5000 tweets. To have a clear understanding of the long-tail distribution, we break down the number of hashtags in 1−5000 category in Table 5.2 (right).

Since tweets that have the same hashtag would mean that they are on the same topic, we build a new collection by combining all tweets with the same hashtag to form a larger document. We then estimate the translation probabilities using Equation 4.4 in this larger collection (hashtag collection). The reason behind this idea is that hashtags help to connect remotely related words once they are in the same document. We then combine the translation probabilities estimated based on the hashtag collection and the original collection based on the following formula:

$$P(w|u) = (1-\gamma)P_C(w|u) + \gamma P_H(w|u) \tag{5.1}$$

where $P_H(w|u)$ is the estimated translation probability based on the hashtag collection and $P_C(w|u)$ is the estimated translation probability based on the whole collection. γ is the parameter that controls the amount of the translation probabilities for each of these collections which we empirically set it. We call this method TM−CombinedHashTags.

Figure 5.4 shows the sensitivity of the MAP measure to γ parameter. It shows that when $\gamma = 0.3$, the optimal value is gained which means that we want to focus more on the estimation probabilities gained from the whole collection.

Table 5.3 shows that leveraging hashtag information helps improve the performance by bringing remotely connected words together. The table also shows that the proposed method is statically significant using Wilcoxon singed ranked test. A main hypothesis here is that a hashtag will not be able to help if it just occurs in one single tweet or a few tweets as it would not help much in bringing together remotely related words. It means that the improvement that we see has come from the hashtags that do occur in many tweets as the analyses in Table 5.2 shows.

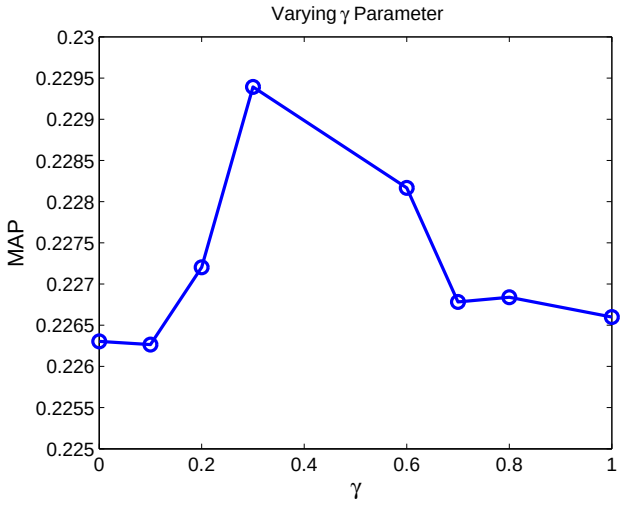

Figure 5.4: Sensitivity of MAP to γ parameter in TM−CombinedHashTags method.

Table 5.3: Comparing the results of TM−CombinedHashTags and CCond. * and + means improvements over BL and CCond are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Method	MAP	P@10
BL	0.21	0.3244
CCond	0.2207*	0.3448
TM−CombinedHashTags	0.2295^{*+}	0.3551

5.3.2 Adaptively Setting the Self-Translation Parameter

This method is a general approach and later we show the results based on news data set (AP90) as well too.

Since a user might use the hashtag word as a query word to indicate the topic he is looking for, we need to adaptively set the self-translation probability for each query word. The first question is: What query terms should have high self-translation probabilities? Intuitively the words that do not rely on translations; because they are already covered well.

The second question is: How we should set this parameter for each query word without relevance judgments? We do approximation and follow the idea of measuring the vocabulary gap in the previous section, and set the self-translation parameter to the percentage of the query word that appear in the top N retrieved documents (pseudo-relevance feedback). The reason is because we assume that top N documents is a good approximation of the relevant documents. We call this method TM−Adaptive$_{QW}$ (adaptive self-translation parameter for each query word).

Table 5.4 shows that adaptively setting the self-translation parameter for two data sets, i.e., Twitter (left) and AP90 (right) further help improves the performance. The table also shows that the proposed

Table 5.4: Comparing the results of TM−Adaptive$_{QW}$ and CCond both based on Twitter data set (left) and AP90 data set(right). * and + means improvements over BL and CCond are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Method	MAP	P@10
BL	0.21	0.3244
CCond	0.2207*	0.3448
TM−Adaptive$_{QW}$	0.23*+	0.3469

Method	MAP	P@10
BL	0.21484	0.3978
CCond	0.264*	0.4234
TM−Adaptive$_{QW}$	0.2698*	0.4234

methods are statically significant using Wilcoxon singed ranked test.

Figure 5.5 shows the sensitivity of MAP measure to the top N retrieved documents. Figure 5.5 (left) shows that when we use top 50 documents to set the self-translation probability on Twitter data set, the optimal value is gained. However for AP90, the optimal value is gained when using top 300 documents as shown in Figure 5.5 (right).

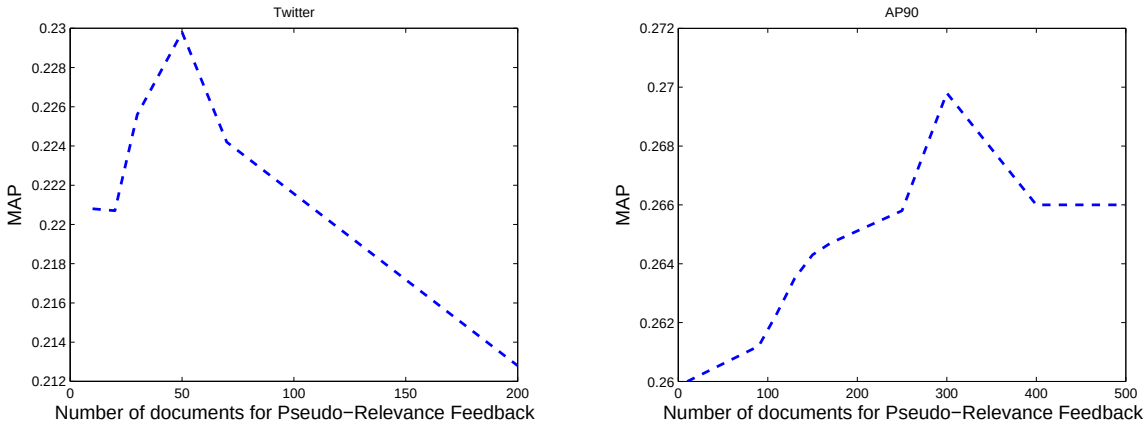

Figure 5.5: Number of documents used for pseudo-relevance feedback to set the self-translation probability for each query word, Left: Twitter data set and Right: AP90 data set.

The third question that we should ask is: How many top-ranked documents should then be used to estimate the self-translation probability? If a query has more relevant documents, we should look at more retrieved documents to set the self-translation probability (i.e., positive correlation). Figure 5.6 confirms that. Figure 5.6 (left) shows positive correlation between the optimal number of documents to set the self-translation parameter and the total number of relevant documents for each query. To aid exposition, in Figure 5.6 (right), we show the correlation between the average number of relevant documents and the optimal number of retrieved documents which again confirms that the more relevant documents we have for each query, the more top N retrieved documents should be used to set the self-translation probability.

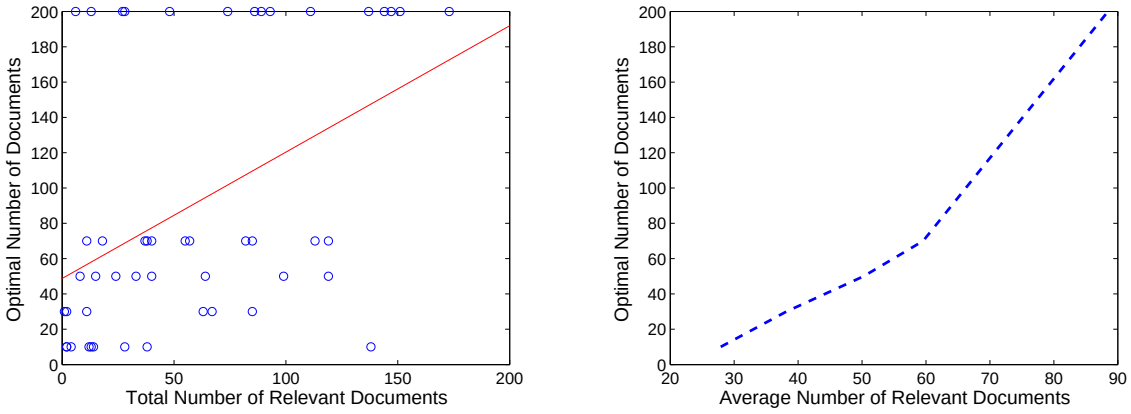

Figure 5.6: Positive correlations between the relevant documents and the top N retrieved documents for each query to set the self-translation probability.

Table 5.5: Comparing the results of TM−Adaptive−IDF and TM−Adptive$_{QW}$ on Twitter data set. * means improvements over BL is statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for MAP measure.

Method	MAP	P@10
Baseline	0.21	0.3244
TM−Adaptive$_{QW}$	0.23*	0.3469
TM−Adaptive−IDF	0.233*	0.3469

Analyzing Constant Self-Translation Probability Constraint:

In Section 4.1, we described that:

In order to have a reasonable retrieval behavior, for all translation language models, we have $p(w|w) = p(v|v), \forall w, v$.

But in this section, we described how to adaptively set a self-translation probability for each query word. In order to obey the "constant self-translation probability", we set the self-translation probabilities for all the query words with the same document frequency to the average of their self-translation probabilities. This is to ensure that the self-translation probabilities for query words with the same document frequency are the same. We expect this method to improve the performance. We call this method "TM-Adaptive-IDF". The results in Table 5.5 indeed indicate that the constant self-translation probability further improves the performance. The reason for not being statistically significant is that there are not many query terms with the same document frequency in this Twitter data set. Figure 5.7 also shows that the absolute MAP difference for these two methods for each query. As shown in the figure, the method TM−Apdative−IDF improves over the TM−Adaptive$_{QW}$.

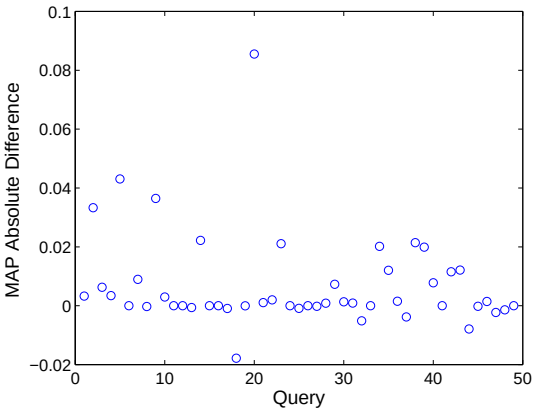

Figure 5.7: Absolute difference of method TM−Adaptive−IDF and TM−Adaptive$_{QW}$.

5.4 Chapter Summary

In this chapter, we studied the direct application of standard translation language model for Twitter search. Further analyses revealed that statistical translation language model can have two benefits; it helps to bridge the vocabulary gap and in case when there is no vocabulary gap it still helps by improving the estimate of term frequency especially for short tweets where the words appear only once and there is no discrimination between the words. We then proposed two ways to improve the performance of statistical translation language model further. The first method is to leverage the hashtag information to bridge the vocabulary gap between remotely related words. The second way is to have a per-term self-translation probability. We evaluated the proposed methods on Twitter data set. Experimental results showed that our proposed solutions are effective, thus improving the accuracy of search results for Twitter search.

For future, it would be interesting to look into some other ways to construct the pseudo documents than to concatenate all tweets containing the same hashtag and see how the estimated translation language model would be different. One alternative way would be to consider different Twitter characteristics including time, users and etc. For example, it would be interesting to see how the estimated translation probabilities based on combining tweets with the same time (or the same user) would change the performance. This strategy might be complementary with the "deficiency" of hashtags most of which only occur in a few tweets.

So far, in the last three chapters, we proposed novel methods and an application of the models for Twitter search that help improve the accuracy of retrieval models in the pre-retrieval stage. In the next chapter, we discuss about how the accuracy of search engines can be improved when we present the results to the user.

Table 5.6: Sample tweets' rank for BL method (top) and CCond method (bottom)

Rank	**Tweet**
1	Get the latest **NSA** job opportunities delivered directly to your phone through the **NSA** Career Links app Available for iPhone and Android
2	**nsa** made me stupid
3	**NSA** Security on Gizmodo
4	Does Google have secret relationship with **NSA** (Relevant document)
5	awww i wanted to watch **nsa**
6	**nsa** jamba tmrw Awesome
7	Alcohol Drugs **NSA** sex
8	**nsa** bhay p0 L0L p
9	**nsa** showtime po kau today
10	FWB works better than **NSA** P

Rank	**Tweet**
1	Get the latest *NSA* job opportunities delivered directly to your phone through the *NSA Career* Links app Available for *iPhone* and *Android*
2	*Watchdog Group questions Google's relationship* with *NSA* Network (Relevant document)
3	*Watchdog Group questions Google's relationship* with *NSA* via NetworkWorld com Community (Relevant document)
4	Some creepy stuff here *Google* and the US Government including *NSA* new report out from Consumer *Watchdog*
5	*nsa showtime po* kau today
6	*nsa showtime* k n b 2m
7	*Spy agency wants* video game to teach spooks to think straight *security CIA FBI NSA* abcdefg
8	*Apple hires former NSA Navy analyst* as *security czar*
9	*NSA Security* on Gizmodo
10	Does *Google* have *secret relationship* with *NSA* (Relevant document)

Chapter 6

Post-Retrieval: Optimization Framework for Negative Feedback

In the previous chapters, we discussed about the pre-retrieval stage (i.e., how the queries are matched with the documents). In this chapter, we discuss what happens after we show the results to the user which is called *post-retrieval* stage.

When a query is so difficult that a large number of top-ranked documents are non-relevant, a user would have to either reformulate the query or go far down on the ranked list to examine more documents, both may decrease the user satisfaction. As a result, improving the effectiveness of search results for such difficult queries would bring user satisfaction which is the ultimate goal of search engines.

A commonly used strategy to improve search results is through feedback techniques, including relevance feedback (e.g., [115, 118, 121]), pseudo-relevance feedback (e.g., [5, 11, 155]) and implicit feedback [129]. In the case of difficult queries, if we can perform effective negative feedback when a user could not find any relevant document on the first page of the search results, we would be able to improve the ranking of the unseen results in the next few pages. It is clear that in this case of negative relevance feedback, we only have negative (i.e., non-relevant) documents since a query is difficult that none of the top-ranked documents are relevant. When a user is unable to reformulate an effective query (which happens often in informational queries due to insufficient knowledge about the relevant documents), negative feedback can be quite beneficial, and the benefit can be achieved without requiring extra effort from users (e.g., by assuming the skipped documents by a user to be non-relevant).

While relevance feedback has been studied extensively, negative feedback has just attracted attention recently. In [148, 149], the authors studied different methods for negative feedback in both language models and vector space model and concluded that negative feedback for language modeling approaches works better than the vector space model. Negative feedback works by excluding documents that are similar to an example negative document. Previous work [149] has shown MultiNeg strategy is most useful when each individual negative document is considered independently, suggesting that negative documents are distracting in different ways. Thus, as training examples, negative examples are sparse. Intuitively, if we can learn from each single negative example to prune aggressively a lot of non-relevant documents from

the top-ranked documents, we should improve the performance more, but in reality there is a risk of over-generalization of a negative example. Thus, an important, yet difficult question is how to appropriately generalize a negative language model; specifically, there are two technical challenges to be solved:

1. What does a general language model mean? How do we formally define generality?
2. Among all the general negative language models of a given negative example, which one should we choose so that we can both maximize its pruning power and avoid over-generalization?

To address the first research question, we propose a formal definition of *generality of a language model* to measure if one negative language model is more general than another. For example, if the query is "jaguar" and the user is looking for documents about jaguar animals, a document containing a jaguar car would be a non-relevant example. This document, however, may not mention the word "car" that often, so if we construct a negative document language model based on this document, the high probability words may be words of a particular jaguar car model such as "Alezon" and "Oxford models". While it is safe to use such words to prune non-relevant documents, their pruning power is limited. A more common word like "car" would be able to prune non-relevant documents more effectively. This generalized negative language model is meant to capture this intuition, and it can be expected to be more effective than the original negative document language model in removing other unseen non-relevant documents, thus improving the accuracy of search results.

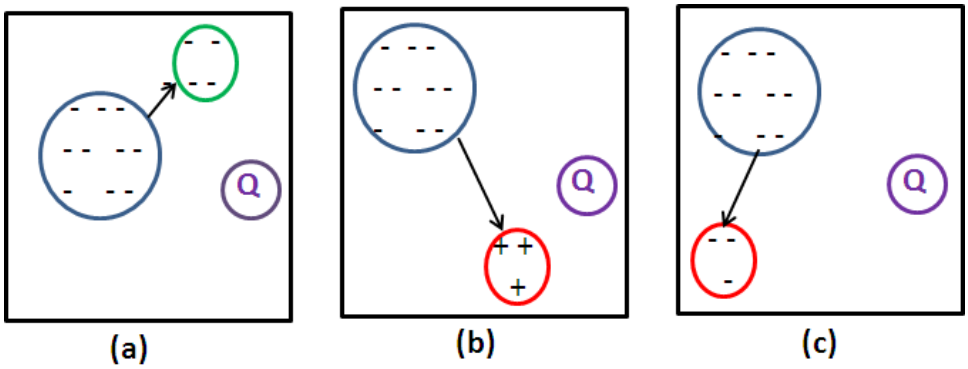

Figure 6.1: An illustrative example. Only case (a) is desirable.

To address the second research question, we propose an optimization framework where we seek a generalized negative language model that optimizes three criteria:

1) closeness to the original negative language model, **2)** not too close to the query and not too far away from the query (if it is too close to the query, there would be a danger of pruning relevant documents, while if it is too far away from the query, it would not be very effective in pruning top-ranked non-relevant

documents), and **3)** a generalization constraint that defines the amount of generalization we would like to achieve. The reason why all these three components are important can be explained in Figure 6.1, where (a) shows that the general negative language model (i.e., green circle) is safe since it is both close to the original negative language model (thus ensures that the pruned documents to be non-relevant) and reasonably close (not too close) to the query (thus can make a difference in the top-ranked results through pruning). Figure 6.1 (b) shows that if we move too far away from the original negative language model and very close to the query, there would be a danger that we would move into the relevant documents zone (red circle). Figure 6.1 (c) shows another undesirable case where we move too far away from the query and as a result, pruning would only affect the lowly ranked documents, and thus is less effective; specifically, although those documents (shown in red circle) are negative documents, removing those documents does not help improving the retrieval accuracy for the given query since they are not in the query zone. So, it is very important to ensure the generalized negative language model to be close (though not too close) to *the given query* and close to *the original negative language model*.

Since the space of all possible negative language models is infinite, we propose two instantiations of our proposed optimization framework to search in a finite space of all feasible language models, which is more tractable.

The first instantiation is based on the KL-divergence distance function where we propose two different methods:

1) Perturbation: to remove the terms (based on the generality measure) in the original negative language model that have less power in pruning non-relevant documents.

2) K-nearest neighborhood (KNN): to search in the neighborhood of the original negative language model to find a more general negative language model.

The second instantiation is based on selecting only powerful terms from the original negative language model where we define an optimization formulation and find an exact solution by solving the optimization formulation directly.

Before we talk about the optimization framework, we give the necessary background for feedback models in language modeling framework.

6.1 Negative Feedback for Language Models

The basic idea in relevance feedback is to extract useful terms from positive documents and use them to expand the original query. When a query is difficult, it is often impossible to obtain positive (or relevant)

documents for feedback. Therefore, the best way would be to exploit the negative documents to perform negative feedback [149]. The idea of negative feedback is to identify distracting non-relevant documents and penalize unseen documents containing such information. More formally:

Given a query q and a document collection $\mathcal{C}$, a retrieval system returns a ranked list of documents $\mathcal{L}$ where l_i is the $i-$th ranked document in the ranked list $\mathcal{L}$. We assume that the query q is difficult so that the top f ranked documents (seen so far by the user) are non-relevant. The goal is how to use these negative examples, i.e., $\mathcal{N} = \{l_1, \ldots, l_f\}$, to re-rank the next r unseen documents in the original ranked list: $\mathcal{U} = \{l_{f+1}, \cdots, l_{f+r}\}$.

The two negative feedback methods proposed in [149] are *SingleNeg* and *MultiNeg* methods which we briefly describe below:

SingleNeg: This method adjusts the original relevance score of a document with a single negative model. Let θ_q and θ_d be estimated query model and document model, respectively. Let θ_N be a negative topic model estimated based on negative feedback documents $\mathcal{N} = \{l_1, \ldots, l_f\}$. The new scoring according to this model is:

$$S(q, d) = -D(\theta_q||\theta_d) + \beta.D(\theta_N||\theta_d) \tag{6.1}$$

In order to estimate θ_N, it is assumed that all non-relevant documents are generated from a mixture model of a unigram language model θ_N and a background language model (generating common words). The log-likelihood of these documents is:

$$L(\mathcal{N}|\theta_N) = \sum_{d \in \mathcal{N}} \sum_{w \in d} c(w, d) log[(1 - \lambda)p(w|\theta_N) + \lambda p(w|\mathcal{C})] \tag{6.2}$$

Where λ (=0.9 in our experiments) is a mixture parameter that controls the weight of the background model, i.e., $p(w|\mathcal{C}) = \frac{c(w,\mathcal{C})}{\sum_w c(w,\mathcal{C})}$. The standard EM algorithm is used to estimate parameters $p(w|\theta_N)$.

MultiNeg: This method adjusts the original relevance scores with multiple negative models. Document d w.r.t query q is scored as follows:

$$\begin{aligned} S(q, d) &= S_R(q, d) - \beta \times S(q_{neg}, d) \\ &= S_R(q, d) - \beta \times \max(\bigcup_{i=1}^{f} \{S(q^i_{neg}, d)\}) \end{aligned} \tag{6.3}$$

$$= \quad -D(\theta_q||\theta_d) + \beta \times \min(\bigcup_{i=1}^{f} \{D(\theta_i||\theta_d)\}).$$

where q_{neg} is a negative query representation and β is a parameter that controls the influence of negative feedback. EM algorithm is used to estimate a negative model θ_i for each individual negative document l_i in $\mathcal{N}$. Then we obtain f negative models and combine them with the above formula in our experiments.

Improving negative document language model: A basic component in both SingleNeg and MultiNeg is a negative document language model (i.e., θ_N in SingleNeg and θ_i in MultiNeg), and the accuracy of the estimate of these negative document models may affect the effectiveness of the negative feedback significantly. A main goal of our study in this chapter is to improve the estimate of a negative document language model through generalizing a basic negative document language model (estimated with an existing approach) with an optimization framework. The improved negative document language models can then be directly plugged into an existing negative feedback method to replace a current negative document language model. In the next section, we present an optimization framework for improving the estimate of negative document language models.

6.2 An Optimization Framework for Generalizing Negative Language Models

6.2.1 Problem Formulation

The goal of our study is to use top f negative examples, i.e., $\mathcal{N} = \{l_1, \ldots, l_f\}$ (with language model θ_N) to build a set of more general negative language models, each corresponding to a negative example, so that these general negative language models are better able to describe other unseen negative documents and improve the ranking of documents by pushing down negative documents in the ranked list. More formally, given

$\theta_N = \{\theta_1, \ldots, \theta_f\}$ which is the original negative language model based on documents in $\mathcal{N}$, where $\theta_1 = \{w_1 : p_1, \ldots, w_m : p_w\}$ (similarly for $\{\theta_2, \ldots, \theta_f\}$), i.e., each language model consists of words along with their probabilities.

Our goal is to estimate $\theta_{G_N} = \{\theta_{G_1}, \ldots, \theta_{Gf}\}$, a set of more general negative language models than θ_N, where each negative language model θ_{G_i} is more general than its corresponding negative language model, θ_i. The improved negative language models θ_{G_N} can then be plugged into a negative feedback method to

improve feedback performance.

6.2.2 Optimization Framework

In order to build a more general negative language model, we propose an *abstract optimization framework* that given θ_N, searches in the space of all language models and finds a set of more general negative language models, i.e., θ_{G_N}. Note that since there are so many general language models, we make it tractable by searching in a *finite space* of all feasible solutions, S.

The objective function is defined as:

$$\theta_{G_N^*} = arg \min_{\theta_{G_N} \in S} (\alpha\delta(\theta_{G_N}, \theta_N) + (1-\alpha)\delta'(\theta_{G_N}, q)) \tag{6.4}$$

Subject to:

$$\mathcal{G}(\theta_{G_N^*}) > \mathcal{G}(\theta_N) + \epsilon \tag{6.5}$$

This abstract optimization framework defines that we would like to search in the finite space of all language models S, to find a more general negative language model $\theta_{G_N^*}$, that is **1)** close to the original negative language model θ_N (the first term), **2)** and close to query q (the second term). The closeness to the query ensures the *pruning* power, and the closeness to the original negative language model both ensures that the feedback model is indeed *accurate* and prevents $\theta_{G_N^*}$ from being too close to the original query. The generalization constraint is to avoid *over-generalization*. So, we want the general negative language model to deviate by *only* ϵ from its original negative language model (i.e., $\mathcal{G}(\theta_{G_N^*}) > \mathcal{G}(\theta_N) + \epsilon$).

δ and δ' are distance functions. $\mathcal{G}(\theta)$ is a generality measure defined in the next section. α is a tradeoff between closeness to the query and closeness to the original negative language model. ϵ controls the deviation from the original negative language model. These parameters can be optimized based on training data (i.e., cross validation) as done in our experiments.

In order to use the proposed optimization framework for negative feedback, there are three remaining problems to be solved:

1. Definition of the generality measure, i.e., $\mathcal{G}(\theta)$.
2. Definition of the similarity/distance functions δ and δ'.
3. Definition of a search algorithm to efficiently enumerate the most promising candidate language models.

Note that the space of potential language models is infinite, but in order to do effective enumeration, our proposed methods search in the finite space (we discuss how to search in the finite space in the next section).

In the next section, we discuss how we solve these problems.

6.3 Instantiation of the Optimization Framework

6.3.1 Generalization of Language Models

In this section, we propose to quantify the generality of a language model by introducing a new notion called *Generality of a language model* to measure if one negative language model is more general than another.

The **Generality measure** is defined as the *expected number* of documents that hit a word. More Formally:

Generality $\mathcal{G}(\theta)$: A language model θ_{G_K} is more general than language model θ_K iff $\mathcal{G}(\theta_{G_K}) > \mathcal{G}(\theta_K)$ where $\mathcal{G}$ is defined as:

$$\mathcal{G}(\theta) = \sum_{w \in \theta} df(w) \times p(w|\theta) \tag{6.6}$$

Where $df(w)$ is the number of documents containing word w in collection $\mathcal{C}$ (document frequency) and $p(w|\theta)$ is the probability of word w given language model θ. Intuitively, the generality in this formula is captured through both the probability of the word in the language model and the number of documents containing that word in the collection.

Next, we define the distance functions, δ and δ' in our optimization framework, and discuss how to efficiently solve the optimization problem.

6.3.2 Distance Functions δ and δ'

In this section, we define two distance functions based on KL-divergence [29] and term-based similarity, respectively. The search strategies vary according to the distance functions.

- **KL-divergence**

  The first instantiation of the abstract optimization framework is to define the distance functions based on KL-divergence. Formally:

$$\delta(\theta_{G_N}, \theta_N) = \frac{1}{2}\left[D(\theta_{G_N}||\theta_N) + D(\theta_N||\theta_{G_N})\right] \tag{6.7}$$

$$\delta'(\theta_{G_N}, q) = D(\theta_{G_N}||\theta_q) \tag{6.8}$$

where θ_q is the query language model. Since there might be some terms which are absent in each of those distributions, we use the symmetric version of KL-divergence for the similarity between two distributions θ_N and θ_{G_N}, (i.e. $\delta(\theta_{G_N}, \theta_N)$).

With these instantiations, our objective function is completely defined. So the next challenge is how to efficiently search in the space S to find an optimal solution. Here we propose two strategies:

1. **Perturbation of θ_N**: For each θ_i in the original negative language model, we build a more general negative language model by removing those words w that satisfy $p(w|\theta_i) \times df(w) < \Psi$ and we still ensure that the final negative language model, i.e., θ_{G_i} is still more general than θ_i, i.e., satisfying the generalization constraint and minimizing the objective function. Note that the probabilities are re-normalized to ensure they are comparable.
2. **K-nearest Neighborhood (KNN)**: For this method, we search among the K-neighbors of the negative language model θ_i for those satisfying the generality constraint, and then among the satisfied ones, we select the best one that minimizes our abstract objective function, i.e., it is both close to the original negative language model and query. Then we use the negative language model of that neighbor instead of the original negative language model as a more general negative language model. We denote this as KNN in our experiments.

- **Term-based similarity**

  In this section, we present another instantiation of our abstract optimization framework where we seek an exact solution in a finite space of all language models defined based on term similarity and selection. This idea is to convert the problem of searching for θ_{G_N} to search for optimal values for binary variables x_i's that would tell us which words should be selected in the generalized negative language model. Specifically, for each top non-relevant document, we get m non-relevant words gained from MultiNeg strategy and feed those terms to an optimization problem using an objective function instantiated as follows.

  In the abstract optimization framework defined earlier, there are two important components, which we now discuss how to instantiate using term-based similarity. For convenience, in the following we

use δ and δ' to denote similarity instead of distance as in the original optimization framework.

1) Generalization of the negative language model: We instantiate this component as:

$$\delta(\theta_{G_N}, \theta_N) = (DF.P)^T \times X \tag{6.9}$$

where

$DF = [df(w_1), \ldots, df(w_m)]^T$ is a vector of document frequencies (in the collection) for each word w_i (document frequency for word w_i).
$P = [p(w_1|\theta_i), \ldots, p(w_m|\theta_i)]^T$ where i is between zero and 10 for each of the 10 non-relevant documents (e.g., $p(w_1|\theta_i)$ is the probability of word w_1 given the language model θ_i).
$X = [x_1, \ldots, x_m]^T$ is the solution vector, which tells which of the words among m words should be included as final words according to the objective function.

The operator "." means element-by-element multiplications of two vectors DF and P.

Maximizing this similarity function has a mixed effect of both staying close to the negative language model (by preferring terms with high probabilities according to the negative language model) and maximizing generalization. As a result, with this formulation, there is no need to have the generalization constraint (i.e., inequality 6.5).

2) Closeness to query:

$$\delta'(\theta_{G_N}, q) = Sim_q^T \times X \tag{6.10}$$

where $Sim_q = [Sim(w_1, q), \ldots, Sim(w_m, q)]^T$ which is the similarity between each word and query word. In case a query consists of multiple words, we get their average similarity. We define the co-occurrence between two words as their similarity. The co-occurrence similarity is based on mutual information as described in Equation 3.13.

According to this instantiation strategy, the optimization problem becomes:

$$\max \left[\alpha(DF.P)^T \times X + (1-\alpha) Sim_q^T \times X\right] \tag{6.11}$$

However, there are two problems with the above formula: **1)** The two components defined in this

optimization framework are not really comparable, so the interpolation parameter α is not very meaningful; **2)** Since we maximize generalization in the first term, we no longer have protection against being too close to the query. To address these two issues, we reparameterize the objective function as follows, where $\gamma < 0$ is a parameter to control the closeness to the query.

Maximize

$$((DF.P)^T \times X + \gamma Sim_q^T \times X)$$

Subject to:

C1:$x_i \in \{0, 1\}$

Figure 6.2: Optimization formulation

This optimization framework has an analytical solution, i.e., $x_i = 1$ (i.e., we would select w_i), iff $df(w_i)p(w_i|\theta_N) > -\gamma Sim(w_i, q)$. The intuition here is that the closer a term is to the query, the higher we would set its threshold of generality; this way, if a selected term is very close to the query, it would have to be a very general term, thus unlikely hurting relevant documents. Note that when $\gamma \geq 0$, the solution is to select all terms with no generalization.

Our solution, i.e., X would tell us which terms should be added to the final negative language model, and we can then recover the θ_{G_N} based on X as follows (re-normalization is done):

$$p(w_i|\theta_{G_N}) \propto x_i p(w_i|\theta_N) \tag{6.12}$$

We denote this method as OptMultiNeg in our experimental results.

6.4 Experiment Design

6.4.1 Data Sets

We experiment with two data sets that are representative of heterogeneous and homogeneous data sets, respectively. Our first data set is ROBUST track of TREC 2004 which has 528,155 news articles [146]. On average, each document has 521.89 words and there are 249 queries[1] in this set. The robust track is a standard ad hoc retrieval with an emphasis on the overall reliability of IR systems which contains difficult queries and is a **heterogeneous** data set. The data set is called "ROBUST".

[1]One query was dropped because the evaluators did not provide any relevant documents for it.

The second data set is the AP88-90 in ad hoc retrieval which is a **homogeneous** data set. It contains 242,918 documents. On average, each document has about 464.226 terms. We used queries $51-200$ for our experiments. The data set is called "AP88-90".

For both data sets, preprocessing of documents and queries is minimum and involves only stemming with Porter stemmer but without removing any stopwords. As in some previous work (e.g., [165]), we did not remove stop words for two reasons: (1) A robust model should be able to discount the stop words. (2) Removing stop words would introduce one extra parameter (e.g. the number of stop words) into our experiments.

Since our goal is to study difficult queries, we consider naturally difficult queries from our data sets as follows:

We follow the definition of naturally difficult queries as in [149]. A query is naturally difficult when its P@10=0 given a retrieval model. For language model (LM), we use the standard ranking function (i.e., KL-divergence retrieval model with Dirichlet prior smoothing [78]) to select their naturally difficult queries. We first optimize the parameter μ (Dirichlet prior) for LM on all data sets. The optimal is gained when $\mu = 2000$ for ROBUST data set and $\mu = 3000$ for AP88-90 data set using Lemur toolkit. We then fix these parameters in all the following experiments. Using the optimal parameter setting, we select those queries whose P@10=0 as our naturally difficult queries. 26 queries in ROBUST and 38 queries from AP88-90 are selected as naturally difficult queries and we experiment with these query sets in the rest of this chapter.

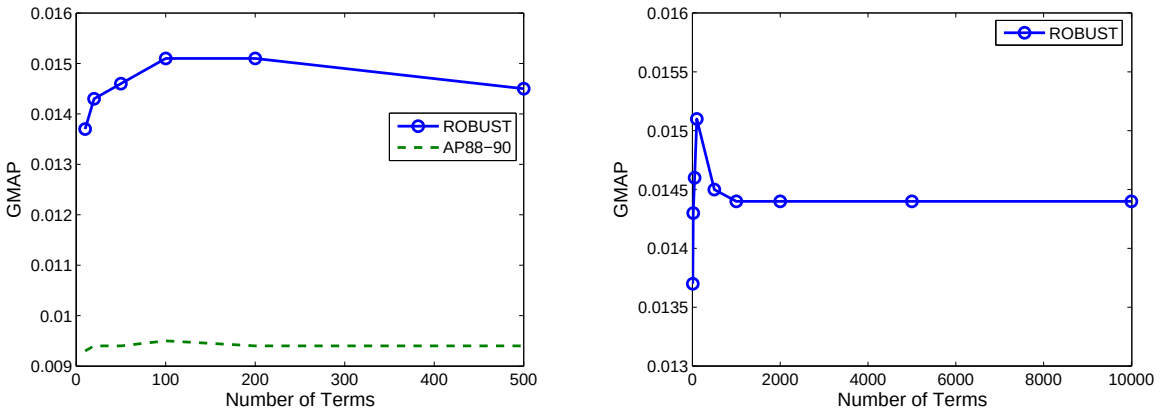

Figure 6.3: Sensitivity of the GMAP measure to the number of terms used for negative feedback with MultiNeg method.

6.4.2 Baselines and Experiment Procedure

Since previous work has shown MultiNeg is the most effective negative feedback method [149], we focus on applying the proposed language model generalization method to MultiNeg, and compare our proposed methods (i.e., OptMultiNeg, KNN and Perturbation) with two state of the art methods, i.e., SingleNeg and MultiNeg. All our proposed solutions follow the same scoring as described in Section 6.1, i.e., each negative document is used to penalize unseen documents, however the language models in our methods are more general.

Our experiment setup is the same as the one adopted in [149]. The goal is to simulate a scenario when a user has found the top-K ranked documents are non-relevant (i.e., these document were skipped by a user without being viewed) and is about to view the rest of the search results. At this point, we can naturally apply negative feedback to re-rank all the *unseen* documents. We set $K = 10$, which simulates the scenario of applying negative feedback when a user has not found any relevant document on the first page of search results and is about to view the next page of results (a realistic assumption in the case of difficult topics).

With this setup, the top-ranked 1000 **unseen** documents for all runs were compared in terms of two sets of performance measures: (1) Mean Average Precision (MAP) and Geometric mean Average Precision (GMAP), which serve as good measures for the overall ranking accuracy. (2) Mean Reciprocal Rank (MRR) and Precision@10 (P@10), which reflect the utility from users perspective who only read the top-ranked documents. Please note that since we are working with difficult queries, GMAP is considered as our **main measure**, however, we show our experimental results based on all measures for the sake of completeness.

In order to set two baseline parameters (i.e., β and ρ), we do cross validation as follows: We fix the number of feedback terms to 100 and learn two baseline parameters, i.e., β (i.e., a parameter to control the influence of the negative feedback) and "number of documents to penalize" (ρ in [149]) based on the training data. Since there are not so many naturally difficult queries in TREC data sets, we do leave-one-out cross validation to learn the parameters for the baselines. The parameters of our proposed methods, i.e., γ, α, Ψ and ϵ are also leaned through leave-one-out cross validation. Thus, the parameters of all the methods (i.e., our proposed methods and baseline methods) are optimized in the same way (i.e., leave-one-out cross validation) to have a fair comparison. Please note that the number of feedback terms is chosen to be 100 (for each document) without loss of generality. As shown in Figure 6.3, when the number of feedback terms are small (smaller than 100), the performance is not

Table 6.1: Performance of the optimization framework on ROBUST data set based on cross validation (top) and upper bound (bottom), * and + means improvements over MultiNeg and SingleNeg are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for GMAP measure since this is our main measure given that we are improving the performance of difficult queries. These results are only based on 100 words extracted from each top non-relevant document.

Methods	MAP	**GMAP**	MRR	P@10
SingleNeg (baseline)	0.0321	0.0134	0.1775	0.088
MultiNeg (baseline)	0.0318	0.0132	0.2124	0.08
KNN	0.0322	0.0138^{*+}	0.2137	0.084
Perturbation	0.0327	0.0136^{*}	0.2597	0.088
OptMultiNeg	0.0365	$\mathbf{0.0144}^{*+}$	0.2804	0.084
OptMultiNeg/MultiNeg	14.7%	9%	32%	5%
OptMultiNeg/SingleNeg	13.7%	7%	57%	-4.5%
Perturbation/MultiNeg	2.8%	3%	22%	10%
Perturbation/SingleNeg	1.8%	1.5%	46%	0%
KNN/MultiNeg	1.2%	4.5%	0.6%	5%
KNN/SingleNeg	0.31%	3%	20%	-4.5%

Methods	MAP	**GMAP**	MRR	P@10
SingleNeg (baseline)	0.0351	0.0145	0.2195	0.092
MultiNeg (baseline)	0.0361	0.0151	0.2388	0.088
KNN	0.034	0.0154^{*+}	0.2445	0.088
Perturbation	0.0393	$\mathbf{0.0159}^{*+}$	0.3407	0.1
OptMultiNeg	0.0378	$\mathbf{0.0156}^{*+}$	0.3223	0.088

good (with MultiNeg baseline method) and when the number of feedback terms are very large (larger than 200), the performance drops. To aid exposition, we increase the number of feedback terms to even 10000 in Figure 6.3 (right), and as shown, the performance drops when the number of feedback terms increases.

We also experiment to get the optimal performance on test queries. For that, we vary β from 0.1 to 0.9 and ρ from 50 to 1000 on test queries and select the best-performing set of parameters (i.e., one β and one ρ for all test queries) according to GMAP measure as done in [149][2] but our main focus would be on the results gained from cross validation. We also vary γ, α, Ψ and ϵ in our proposed solutions as shown in Figures 6.4 and 6.5 to get the optimal parameters.

6.5 Experimental Results

6.5.1 Effectiveness of our Proposed Solutions

In order to see the effectiveness of our proposed solutions, we compare them with the baselines methods.

[2]The authors only reported the upper bound results (or the optimal results) without reporting the results based on cross validation. However, one should note that parameters should be learned based on training data sets.

Table 6.2: Performance of the optimization framework on AP88-90 data set based on cross validation (top) and upper bound (bottom), * and + means improvements over MultiNeg and SingleNeg are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for GMAP measure since this is our main measure given that we are improving the performance of difficult queries. These results are only based on 100 words extracted from each top non-relevant document.

Methods	MAP	**GMAP**	MRR	P@10
SingleNeg (baseline)	0.0377	0.0091	0.1500	0.0631
MultiNeg (baseline)	0.0389	0.0093	0.1521	0.0651
KNN	0.0388	0.0094^{+}	0.1554	0.0657
Perturbation	0.0386	0.0093^{+}	0.1523	0.0631
OptMultiNeg	0.0391	$\mathbf{0.0096}^{*+}$	0.1937	0.0658
OptMultiNeg/MultiNeg	0.5%	3%	27%	1.1%
OptMultiNeg/SingleNeg	3.7%	5.5%	29%	4%
Perturbation/MultiNeg	-0.7%	0%	0.13%	-3%
Perturbation/SingleNeg	2.4%	2%	1.5%	0%
KNN/MultiNeg	-0.25%	1%	2.2%	0.9%
KNN/SingleNeg	2.9%	3%	3.6%	4.1%

Methods	MAP	**GMAP**	MRR	P@10
SingleNeg (baseline)	0.0381	0.0092	0.1517	0.0684
MultiNeg (baseline)	0.0396	0.0094	0.1914	0.0684
KNN	0.0393	0.0095^{+}	0.1755	0.0684
Perturbation	0.0403	0.0096^{*+}	0.197	0.071
OptMultiNeg	0.0392	$\mathbf{0.0097}^{*+}$	0.1950	0.0658

The results are shown in Tables 6.1 and 6.2 based on both collections ROBUST and AP88-90, respectively. Table 6.1 (top) shows the cross validation results on ROBUST data set and Table 6.1 (bottom) shows the upper bound results. The results in Table 6.2 (top) and (bottom) also show cross validation and upper bound results based on AP88-90 data set, respectively. The upper bound baseline results are comparable to their corresponding results reported previously [149] (the authors of [149] did not report the cross validation results). From these two tables, we have the following observations:

(1) The results both based on upper bound and cross validation show that our proposed methods outperform the baselines in terms of GMAP (since this serves as our main measure and we maximize based on this measure when learning the parameters). For example, on ROBUST data set, the OptMultiNeg method can improve GMAP from 0.0132 (in MultiNeg method) to 0.0144, about 9% relative improvement which is a significant improvement given that the difficult queries are harder to be improved. On AP88−90, our proposed methods can also significanlty improve over baseline methods. The results are also statistically significant based on Wilcoxon signed-rank tests (for GMAP measure) for those cases marked in the tables. We also show the percentage improvement for our methods over the baselines (based on cross validation only) in Tables 6.1 and 6.2 (top). For example, OptMultiNeg/MultiNeg means the improvement of OptMultiNeg over MultiNeg baseline method. Please note that we only show the percentage improvement for cross

Table 6.3: Performance of the optimization framework on ROBUST data set based on cross validation (top) and Upper bound (bottom), * and + means improvements over MultiNeg and SingleNeg are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for GMAP measure. These results are based on all words extracted from each top non-relevant document.

Methods	MAP	GMAP	MRR	P@10
SingleNeg (baseline)	0.0336	0.0137	0.1861	0.088
MultiNeg (baseline)	0.0287	0.0127	0.1502	0.08
OptMultiNeg	0.0316	**0.0141**$^{*+}$	0.2082	0.08

Methods	MAP	GMAP	MRR	P@10
SingleNeg (baseline)	0.0350	0.0147	0.2119	0.088
MultiNeg (baseline)	0.0318	0.0144	0.2147	0.088
OptMultiNeg	0.0326	**0.0151**$^{*+}$	0.2463	0.088

Table 6.4: Performance of the optimization framework on AP88-90 data set based on cross validation (top) and upper bound (bottom), * and + means improvements over MultiNeg and SingleNeg are statistically significant with Wilcoxon signed-rank test, respectively. We only show the significance tests for GMAP measure. These results are based on all words extracted from each top non-relevant document.

Methods	MAP	GMAP	MRR	P@10
SingleNeg (baseline)	0.0395	0.0094	0.1702	0.0710
MultiNeg (baseline)	0.0386	0.0091	0.1723	0.0684
OptMultiNeg	0.0389	**0.0096**$^{*+}$	0.1869	0.0710

Methods	MAP	GMAP	MRR	P@10
SingleNeg (baseline)	0.0396	0.0095	0.1752	0.071
MultiNeg (baseline)	0.0401	0.0096	0.1981	0.071
OptMultiNeg	0.0396	**0.0097**$^{*+}$	0.1917	0.0763

validation results *only* since these are our main results.

(2) Another interesting observation is that although we only maximize based on GMAP, the results of our proposed methods also outperform MAP and MRR measure in most cases (cross validation results).

(3) Comparing our proposed methods shows that the *OptMultiNeg* method outperforms (according to GMAP measure) all the other proposed methods since it finds an exact optimal solution.

Given that these are all very difficult queries where the state of the art retrieval models worked poorly and negative feedback can be done automatically based on implicit feedback information without requiring any additional user effort, these results are quite encouraging.

Table 6.5 shows some sample words along with their probabilities selected both based on MultiNeg and OptMultiNeg for TREC query 690 (This is an example where the OptMultiNeg method outperformed the baselines in terms of GMAP measure a lot). We only show top 10 selected terms and also calculate their generality measures for these two methods. It is shown that the OptMultiNeg method is more general (i.e., $\mathcal{G}_{10}(\theta') > \mathcal{G}_{10}(\theta)$). Please note that the generality values calculated here is based on only 10 shown words;

we should also mention that $\mathcal{G}_{100}(\theta') > \mathcal{G}_{100}(\theta)$, i.e., our OptMultiNeg is more general than MultiNeg when top 100 words are selected (this is the number of feedback terms that we experimented with). In addition, we show another example (TREC query 343) where the performance was hurt a lot. The sample words selected are shown in Table 6.6 which shows the over-generalization both in terms of the words themselves (e.g., words are not specific compared to the original words) and in terms of the generalization value.

Another experiment setup is to give all the words extracted from the original non-relevant documents to the OptMultiNeg method and let the method choose among them (instead of choosing among top 100 words). Our hypothesis is that the results of our OptMultiNeg from this experiment should also be better than that of baselines for the case when we have only 100 terms. The results of such experiments are shown in Tables 6.3 and 6.4 for both data sets. As shown in the tables, the results are worse than their corresponding results in Tables 6.1 and 6.2 (as we expected from the analyses shown in Figure 6.3). However, the OptMultiNeg still outperforms the baselines which confirms our hypothesis.

Efficiency of our proposed methods: Since we only have 10 negative examples (first-page result simulation), building a language model for each of them can be done efficiently online. Also, since we search in the finite space of all the potential language models, the search is fast enough to be done online, i.e., even for the OptMultiNeg method, search does not take a lot of time to select words since the space of the selection is also finite.

Table 6.5: 10 Sample word selected from MultiNeg model (left) and OptMultiNeg (right) for query **690= "colleg educ advantag"**. Note that words are stemmed.

q	$p(q\|\theta)$
shanghai	0.0512
school	0.0354
we	0.0336
reform	0.0171
teacher	0.0155
system	0.0141
higher	0.0138
establish	0.0125
learn	0.0124
graduat	0.0117
$\mathcal{G}_{10}(\theta) = 32,899$	

q	$p(q\|\theta')$
shanghai	0.275
we	0.1804
reform	0.0921
system	0.0760
establish	0.0672
percent	0.0542
cooper	0.0450
must	0.0444
construct	0.0441
develop	0.0294
$\mathcal{G}_{10}(\theta') = 33,461$	

6.5.2 Parameter Sensitivity Study

Since there are parameters associated with the proposed methods, in this section, we analyze the sensitivity of GMAP to the parameters.

Table 6.6: 10 Sample word selected from MultiNeg model (left) and OptMultiNeg (right) for query **343= "polic death"**. It is an example that over-generalization hurts the performance. Note that words are stemmed.

q	$p(q\|\theta)$
injuri	0.0336
anc	0.0318
attack	0.0264
support	0.0204
hostel	0.0200
incid	0.0193
ifp	0.0180
allegedli	0.0158
secur	0.0141
member	0.0140
$\mathcal{G}_{10}(\theta) = 10,990$	

q	$p(q\|\theta')$
anc	0.1325
support	0.0852
secur	0.0591
member	0.0585
forc	0.0497
were	0.0479
week	0.0402
sub	0.04023
11	0.0323
region	0.0317
$\mathcal{G}_{10}(\theta') = 20,850$	

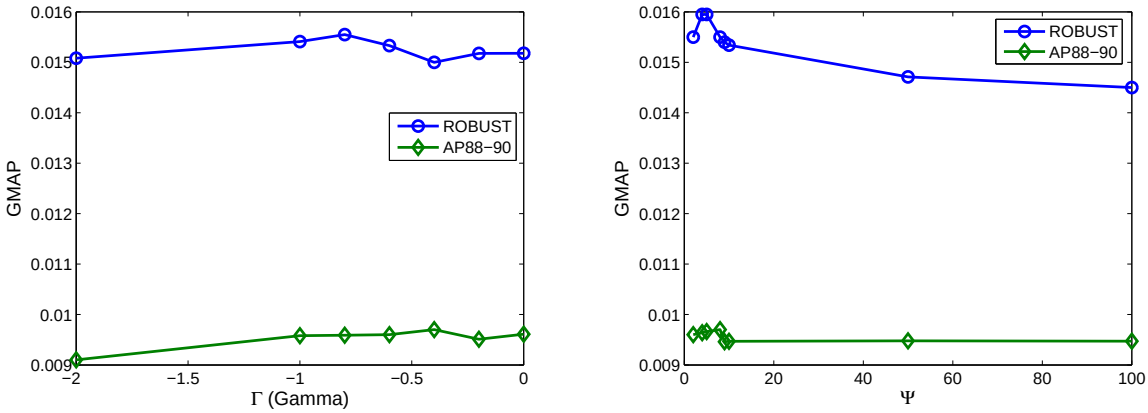

Figure 6.4: Sensitivity of GMAP to different parameters: γ in OptMultiNeg method (left), Ψ in Perturbation method (right).

Figure 6.4 (left) shows sensitivity of GMAP to γ in OptMultiNeg method for both ROBUST and AP88-90 data sets. We first fix β, ρ to their optimal values and vary γ to see how it changes to the GMAP value. As shown in the figure, when $\gamma = 0$, the baseline method (MultiNeg) is achieved, and by choosing an appropriate negative value, the performance can be improved, though when γ is too large, the performance would be poor, likely because of over-generalization.

Figure 6.4 (right) shows the sensitivity of Ψ in Perturbation method. As shown in the figure, when we increase the amount of Ψ, it hurts the performance, so it indicates that over-generalization hurts the performance and Ψ should be set to: $1 < \Psi < 5$ to ensure improvement in the performance.

Figure 6.5 (left) also shows the sensitivity of GMAP measure to parameter α in KNN method[3] which is a tradeoff between closeness to query and closeness to negative language model. The best performance is gained when $0.4 < \alpha < 0.6$ which confirms our hypothesis that the general negative language model should

[3]The same pattern can also be seen with Perturbation method.

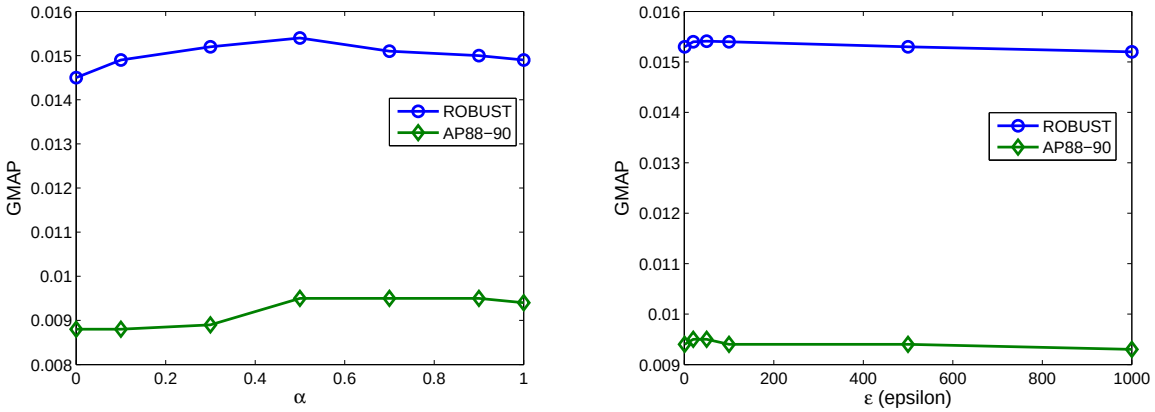

Figure 6.5: Sensitivity of GMAP to different parameters: α in KNN method to GMAP (left), ϵ in KNN method to GMAP (right).

be close to both query and original negative language model.

Figure 6.5 (right) also shows the sensitivity of GMAP measure to parameter ϵ in KNN method. As we increase ϵ, the performance hurts indicating that over-generalization hurts the performance and ϵ should be set to: $1 < \epsilon < 50$.

6.6 Chapter Summary

How to help users when their queries do not work well with the state of the art retrieval methods is a very important, yet difficult challenge. Negative feedback is an important and useful technique for tackling this challenge, and can be done automatically without requiring extra user effort based on implicit feedback information (such as skipping all the results on the first page and attempting to view additional results on the next page).

In this chapter, we addressed the problem of data sparseness in negative feedback in the language modeling framework by proposing an abstract optimization framework, in which we learned from a few top-ranked non-relevant examples, and searched in the space of all candidate language models to build a more general negative language model. This general negative language model has been shown to have more power in pruning the non-relevant documents on two representative TREC data sets, outperforming the state of the art negative feedback methods significantly for difficult queries. The proposed methods are general and can be potentially implemented in any search engine applications to improve a user's experience in the case of difficult topics.

This work is only a first step in the exploration of the general idea of generalization of language models. There are many interesting directions to further explore. First, we can relax the feedback condition to

include a small number of positive feedback examples, which would lead to the interesting problem of how to generalize both negative and positive language models simultaneously. Second, it would be interesting to apply explanation-based learning (EBL) [36] to feedback and construct a generalized language model based on a formal explanation of why a document has been judged as non-relevant (or relevant). Finally, we believe that the idea of generalizing language models can also be potentially applied to many other tasks in interactive retrieval where we need to infer a user's intent based on the user's behavior.

In the next chapter, we talk about the interactive retrieval to interact with the users several times helping them find their relevant information when they have complex information needs.

Chapter 7

Interactive Retrieval: Interactive Relevance Feedback

Difficult queries tend to cause more interactions with the search engine, so it is important to study how to optimize the interactive retrieval which is the topic of this chapter.

After we present the results to the user (i.e., post-retrieval stage), there is a tradeoff between presenting search results with the highest immediate utility to a user (but not necessarily most useful for collecting feedback information) and presenting search results with the best potential for collecting useful feedback information (but not necessarily the most useful documents from a users perspective). Especially when a query is difficult, the interaction with the user is more needed but how to optimize such a tradeoff (called exploration-exploitation tradeoff) is a key to the optimization of the overall utility of feedback to a user in the entire session of user interaction with the system and has not studied before. In this chapter, we address this important problem.

A lot of research has been done on relevance feedback. However, most existing work is focused on developing and improving relevance feedback methods (e.g., [118, 121, 138, 164]), where the goal is to optimize the ranking of results based on a given set of relevance judgments. Typically, the relevance judgments are assumed to be collected on some top-ranked documents that would be shown to a user as initial results. Although this strategy of relevance feedback is very natural as a user can judge documents while viewing search results in a normal way, the judgments collected in this way (i.e., judgments on the top-ranked documents) may not necessarily be the most useful judgments for relevance feedback. As an extreme case, if the top documents all have similar contents, the judgments on all these documents would be clearly redundant, and would not be so useful for relevance feedback as if we collect the same number of judgments on a more diversified set of documents. Clearly, in such an extreme case, the benefit brought by relevance feedback to the user in the next iteration of the interactive retrieval process would be relatively small, at least not as much as the user would have if the judgments were collected on a more diverse set of documents. This shows a deficiency of the traditional (standard) relevance feedback strategy: *it does not attempt to obtain the most useful judgments.*

This observation has motivated some researchers to study active feedback methods [130, 158, 157], where

the goal is to choose documents for relevance feedback so that the system can learn most from the feedback information. The idea of these methods is usually to choose a set of diverse documents for judgments since judgments on diverse documents can be expected to be more informative and thus helpful for the system to learn the user's information need. For example, in [130], the top-ranked documents are first clustered and only one document is selected from each cluster to form a diverse set of documents to present to the user. However, presenting diverse documents to users means that we generally do not present the search results in the order of relevance. Thus, the utility of the presented documents to the user is generally lower than that of presenting the top-ranked documents. This shows that active feedback has a different deficiency: *it does not attempt to optimize the utility of the presented documents to the user.*

These observations illustrate an interesting dilemma in interactive relevance feedback: On the one hand, we want to present the top-ranked documents to a user so that the presented documents would be most useful for the user in this interaction cycle as in standard relevance feedback. However, such a strategy does not generate the most useful judgments for feedback. On the other hand, if we diversify the results in order to obtain more useful judgments, which would bring more utility to the user in the next interaction cycle, we would risk on decreasing the utility to the user in the current interaction cycle. We refer to this dilemma as *exploration-exploitation tradeoff* [75, 139].

Intuitively, if we present top ranked results to the user, it is very useful for the user since there is no discount compromise of the utility, which can be regarded as emphasizing *exploitation* of all the existing information about the user's information need, but it does not help the system to *explore* the space of all the potentially relevant documents to the user. On the other hand, if we show diversified results (i.e., emphasizing *exploration*), there is a concern of decreasing the utility from the user perspective, thus possibly compromising *exploitation*.

Clearly, in an interactive retrieval system, what matters to a user is the *overall* utility of relevance feedback, which means that we need to optimize this exploration-exploration tradeoff so as to optimize the overall utility over an interaction session[1] which includes both the current interaction cycle and the next interaction cycle after feedback. To the best of our knowledge, no existing work has addressed the problem of optimizing the exploration-exploitation tradeoff for relevance feedback. Indeed, the standard relevance feedback work is focused on exploitation without considering exploration, while the active feedback work does the opposite – focused on exploration without considering exploitation; each can be regarded as optimizing feedback for just one interaction cycle. Indeed, this tradeoff problem touches a more general issue of how to optimize the utility of a retrieval system over an interaction session for a user [44].

[1]We define **session** as viewing multiple search results for the same user query through interacting with the system (e.g., clicking on buttons).

In this chapter, we study how to optimize the exploration-exploitation tradeoff in relevance feedback based on the following interactive process: A user is assumed to have a high-recall information need. After issuing a query, the user is assumed to sequentially view results up to N top-ranked results. We further assume that in this process, the user would need to fetch unseen results at least twice due to the limited number of results that can be displayed on one screen, resulting in two natural time points when a system can learn from the feedback information collected so far to optimize the unseen results. One instance of this scenario is that a user views three pages of search results; in such a case, the system can learn from the feedback information collected from the first page of results to optimize the results shown on the second page right before the user navigates into the second page, and can also do the same for the third page when the user reaches it. While most users of a Web search engine do not go this far when viewing results, we would expect them to do so when the search results are displayed on a much smaller screen such as on a smartphone. Also, a user often needs to scroll down on the first page to view all the results; in such case, we may also view the "scrolling" as to fetch more unseen results, thus a point when the system can re-rank the unseen results. For convenience of discussion, we will refer to the very first batch of results seen by the user as the results on the first "page"(or segment), the results seen by the user through either scrolling or clicking on a next-page button as the results on the second "page" (or segment) and so on so forth.

Since more diversified results would mean more exploration and thus less exploitation, while less diversified would mean more exploitation and less exploration, we essentially convert this tradeoff problem to one of optimizing this diversity parameter. We propose and study two methods to optimize this diversity parameter. The first one assumes a fixed optimal value for all the queries (fixed-coefficient diversification), and the second learns a query-specific optimal value for each query (i.e., adaptive diversification).

We evaluate our proposed methods on three representative TREC data sets. The experimental results demonstrate that both of our proposed methods can effectively optimize exploration-exploitation tradeoff and achieve higher utility for the session than both traditional relevance feedback which ignores *exploration* and pure active feedback which ignores *exploitation*. Moreover, the adaptive diversification method is better than the fixed-coefficient diversification method due to its optimization of the tradeoff at a per-query basis.

In the next section, we formally define the problem.

7.1 Problem Formulation

Given a query q and a document collection $\mathcal{C}$, we consider an interactive retrieval system where we assume a user would "progressively" view three segments of results through either scrolling or clicking on next-page button. The number of results in each segment is not necessary equal. For example, S_1 can be the top 5 results seen by a user in the beginning (the screen size does not allow the user to see all the 10 results), S_2 can be the bottom 5 results on the first page, and S_3 can be the next 10 results on the second page. The system first uses the query to retrieve k documents $S_1 = (d_{11}, ..., d_{1k})$ and present them on the first segment S_1. The user is then assumed to click on any relevant documents on this segment, leading to a set of relevance judgments on the first segment, J_1. We assume that the user would continue viewing the second segment of search results, and at least some results on the third segment. We will focus on studying relevance feedback that happens after the user finishes viewing all the results on the second segment.

Our goal is to study how to use all the information collected after the user views the first segment (i.e., judgments J_1) to optimally interact with a user on the second segment so as to optimize the overall utility over the session of the user interacting with all the three segments. Specifically, we would like to optimize the diversification of the results on the second segment $S_2 = (d_{21}, ..., d_{2m})$ so that the relevance feedback that happens just before the user views the results on the third segment would deliver the maximum overall utility on both the second segment and the third segment. (The utility on the first segment has already been fixed in this setup, thus it is irrelevant for the optimization.) As discussed before, more diversification leads to more exploration and less exploitation, thus optimizing the diversification is a way to optimize exploration-exploitation tradeoff.

A common way to generate diversified results is a greedy algorithm called Maximal Marginal Relevance (MMR) [16] which models both topical relevance and redundancy of documents. With this strategy, after we have already selected documents $d_{21}, \ldots, d_{2,i-1}$ for the second page, document d_{2i} would be selected to maximize the following score:

$$S(d_{2i}; d_{21}, \ldots, d_{2,i-1}) = (1 - \lambda) S_R(d_{2i}, q) + \lambda S_{novelty}(d_{2i}; d_{21}, \ldots, d_{2,i-1}) \tag{7.1}$$

where q is the query, $S_R(d_{2i}, q)$ is a relevance-based scoring function (i.e., regular retrieval function). $S_{novelty}$ is the novelty value of d_{2i} w.r.t. $\{d_{21}, \ldots, d_{2,i-1}\}$, and $\lambda \in [0, 1]$ is the diversification parameter that controls the degree of diversification, or equivalently, the amount of exploration.

The results on the third segment $S_3 = (d_{31}, ..., d_{3p})$ would be generated using relevance feedback on the judgments collected on the second segment. Clearly, the utility of this third segment would be affected

by the usefulness of the collected judgments J_2 which further depends on the diversification on the second segment. Our goal is to optimize the diversification coefficient, i.e., λ, to maximize the *overall utility* of the results on the second segment and the third segment, i.e., to optimize the exploration-exploitation tradeoff. Formally, we assume that there is a function A that can map a query q and its corresponding relevance judgments J_1 on the first segment to the optimal diversification coefficient, i.e., $\lambda = A(q, J_1)$. Later, we will propose two different ways to compute $A(q, J_1)$. One way is simply to search for an optimal λ over a set of training queries and take this fixed coefficient as the optimal value for all the unseen test queries. The other way is to use a machine learning method to learn a potentially different optimal λ for each query.

Since our main goal is to study different methods for optimizing the diversification coefficient, in our experiments, we assume $|S_1| = |S_2| = |S_3| = 10$ to simulate a scenario when the user views the first three pages of results and we refer to S_1 as first-page result, to S_2 as second-page result and S_3 as third-page result. The proposed method does not depend on this configuration, though, and can be applied to other scenarios as well.

In the next section, we describe the methods used throughout this chapter in more details.

7.2 Background

We now present all the components, i.e., basic retrieval model, feedback methods and novelty method in problem setup (section 7.1) in more details.

Basic Retrieval Model: We use the Kullback-Leibler divergence retrieval model with Dirichlet prior smoothing as our basic retrieval model for ranking documents on all the three pages (on the second page, it is used together with a novelty measure function to diversify results).

Feedback Model: In [164], authors have defined a two-component mixture model (i.e., a fixed background language model, $p(w|\mathcal{C})$, estimated using the whole collection and unknown topic language model to be estimated) and assumed that the feedback documents are generated using such a mixture model. Formally, let θ_T be the unknown topic model and $\mathcal{F}$ be a set of feedback documents. The log-likelihood function of the mixture model is:

$$\mathcal{L}(\mathcal{F}|\theta_T) = \sum_{d \in \mathcal{F}} \sum_{w \in V} c(w, d) \log[(1 - \alpha)p(w|\theta_T) + \alpha p(w|\mathcal{C})] \tag{7.2}$$

Where $\alpha \in [0, 1)$ is a mixture noise parameter which controls the weight of the background model. EM algorithm can be used to estimate $p(w|\theta_T)$ which is then interpolated with the original query model $p(w|q)$ to obtain an improved estimation of the query model:

$$p(w|\theta_q) = (1 - \gamma)p(w|q) + \gamma p(w|\theta_T) \tag{7.3}$$

Where γ is the feedback coefficient to be set manually.

To perform relevance feedback after the user views the second page, we perform the two-component mixture model for positive feedback and the MultiNeg strategy for negative feedback; both have been shown to be effective for the respective feedback task.

Novelty Measure: To compute the novelty score $S_{novelty}$ described in section 7.1, we use a method proposed in [163]. The best performing novelty measure they reported is MixAvg. The novelty measure is based on a two-component generative mixture model in which one component is the old reference topic θ_0 estimated based on $\{d_{21}, \cdots, d_{2,i-1}\}$ described in section 7.1 and the other is the background language model (e.g., general English model). Specifically, let θ_B be a background language model with a mixing weight of μ, the log-likelihood of a new document $d = \{w_1, \cdots, w_n\}$ is:

$$\mathcal{L}(\mu|d, \theta_0) = \sum_{i=1}^{n} \log[(1 - \mu)p(w_i|\theta_0) + \mu p(w_i|\theta_B)] \tag{7.4}$$

And the estimated novelty score is obtained by:

$$\mu^* = arg \max_{\mu} \mathcal{L}(\lambda|d, \theta_0) \tag{7.5}$$

The EM algorithm can be used to find the unique μ^* that maximizes the score. We call this method MixAvg-MMR in our experiments.

One should note that our focus is studying the parameter λ (novelty parameter) described in section 7.1 and the parameters that discussed in this section, will be fixed to their optimal or default values as suggested in the literature.

7.3 Learning to Optimize Diversification

We consider two ways to optimize the diversification parameter λ. The first is simply to vary this parameter on a training data set and select the best performing value, and set this parameter to such a fixed optimal value across all the test queries. We call this approach fixed-coefficient diversification.

Intuitively, the right amount of diversification may depend on queries (e.g., a query with more subtopics may benefit from more diversification). Thus for our second method, we propose a learning method to "learn when to diversify the results" by adaptively learning this coefficient for each query, which we present it in details in the next section.

7.3.1 Features for Diversification

In this section, we describe a learning approach that adaptively learns the λ parameter for each query.

We first identify some features that are correlated to the diversification parameter, i.e. λ. These features are computed based on *only* the first-page results and the judgments on it, J_1. Because at the time of optimizing the diversification on the second page, we only have this much information. In the next subsection, we will discuss how we combine these features to learn the function $A(q, J_1)$ (which can be used to compute an optimal value for λ for each query) using the past queries as training data. The learned function $A(q, J_1)$ can then be used for future queries to predict new λ. Note that since we do not learn a fixed λ value, but a function for computing an optimal value of λ, this allows us to obtain a potentially different optimal value of λ for each query.

The following notations will be used in the definition of features. F_{rel} and F_{nonRel} are the set of relevant and non-relevant documents in J_1, respectively. $c(w, q)$ is the count of term w in query q. $|q| = \sum_{w \in q} c(w, q)$ is the total number of words in query q. $p(w|\theta_q) = \frac{c(w,q)}{|q|}$ is the query language model. $p(w|\mathcal{C})$ is the collection language model. $p(w|\theta_{F_{rel}})$ and $p(w|\theta_{F_{nonRel}})$ are the language models of relevant documents and non-relevant documents on the first-page results, respectively.

The followings are the features extracted from the first-page results. Please note that the reason for showing the first-page results to the user is to get some limited information about relevant documents and as a result extract the following features which are needed for our learning algorithm.

Query Length: As mentioned in [55], query length affects retrieval performance and is defined as:

$$|q| = \sum_{w \in q} c(w, q) \tag{7.6}$$

Query Distribution: Each term is associated with an inverse document frequency which describes the informative amount that a term in the query carries. It is defined as follows [55]:

$$QDist = \frac{idf_{max}}{idf_{min}} \tag{7.7}$$

Where idf_{max} and idf_{min} are the maximum and minimum idfs among the terms w in query q, respectively.

Query Clarity: Query clarity has been shown to predict query difficulty [30]. When a query is difficult, it might mean that it has different interpretations, so in order to complete the picture of all aspects of the query, we need to provide a diverse set of documents to make sure that all aspects of the query are covered.

(1) According to definition [30], the clarity of a query is the Kullback-Leibler divergence of the query model from the collection model. For our case, the query model is estimated from the relevant documents in F_{rel} which is defined as follows:

$$QClar_1 = \sum_{w \in F_{rel}} p(w|\theta_{F_{rel}}) log_2 \frac{p(w|\theta_{F_{rel}})}{p(w|\mathcal{C})} \tag{7.8}$$

Where $p(w|\theta_{F_{rel}})$ is estimated as $\frac{c(w,F_{rel})}{\sum_{w' \in F_{rel}} c(w',F_{rel})}$, where $c(w, F_{rel})$ is the number of word w in F_{rel} (i.e., relevant documents).

(2) To avoid the expensive computation of query clarity, authors in [55] proposed a simplified clarity score as a comparable pre-retrieval performance predictor. It is calculated as follows:

$$QClar_2 = \sum_{w \in q} p(w|\theta_q) log_2 \frac{p(w|\theta_q)}{p(w|\mathcal{C})} \tag{7.9}$$

Query Entropy[26, 136]: If query entropy is high, it means that the query covers broad topics, as a result, we do not need to diversify the results. We define two query entropies as follows:

(1)

$$QEnt1 = \sum_{w \in q} -p(w|\theta_q) log_2 p(w|\theta_q) \tag{7.10}$$

(2) Since a query is often short, we compute another query entropy score based on the relevant documents in F_{rel} as follows:

$$QEnt2 = \sum_{w \in F_{rel}} -p(w|\theta_{F_{rel}}) log_2 p(w|\theta_{F_{rel}}) \tag{7.11}$$

Where $p(w|\theta_{F_{rel}})$ is estimated as $\frac{c(w,F_{rel})}{\sum_{w' \in F_{rel}} c(w',F_{rel})}$.

Number of Relevant documents: If the query has *high initial precision*, i.e., the fraction of retrieved relevant documents on the first-page results are high, we do not need to diversify the results; as a result we consider the number of relevant documents in the first-page results as a feature:

$$num = |F_{rel}| \tag{7.12}$$

Virtual Mean Average Precision: Another feature to capture *high precision* is to calculate *Average precision* for the first-page results as in [90]:

$$VirMAP = \sum_{d \in F_{rel}} \frac{prec(r_d)}{10} \tag{7.13}$$

Where r_d is the rank of document d and $prec(r_d)$ is the precision of top r_d documents.

Separation of Relevant and Non-Relevant Documents: If the query is *clear* enough (relevant and non-relevant documents are separated) we do not need to diversify the results. Thus we introduce the following two features to measure the separation between relevant documents in slightly different ways.

(1) Jensen-Shannon Divergence (JSD)

$$JSD = \frac{1}{2}[D(\theta_{F_{rel}}||\theta_{F_{nonRel}}) + D(\theta_{F_{nonRel}}||\theta_{F_{rel}})] \tag{7.14}$$

Where D is the Kullback-Leibler divergence between the two models.

(2) Cosine Similarity: We denote the term frequency vectors of F_{rel} and F_{nonRel} as $V_{F_{rel}}$ and $V_{F_{nonRel}}$, respectively. The cosine similarity is then defined as:

$$CosSim = \frac{V_{F_{rel}}.V_{F_{nonRel}}}{||V_{F_{rel}}||.||V_{F_{nonRel}}||} \tag{7.15}$$

Diversification (Div): Intuitively, if the baseline results are already diversified, we do not need to do more diversification; one measure to capture that is as follows: We cluster the top 30 results from Kullback-Leibler divergence retrieval to 5 clusters using K-Means algorithm [37]. We then consider the ratio of the size of the first largest cluster to the second largest cluster. If this ratio is small, it means that we have already formed multiple clusters and the results are already diversified, so we do not need to diversify more.

Analysis of computational efficiency of the features: The features discussed above can be categorized into query-based features and document-based features. Query-based features are those that are calculated based on the query. Examples of these kinds of features are: query length $|q|$, query distribution $QDist$, query entropy $QEnt1$ and etc. Document-based features are those that are calculated based on relevant and non-relevant documents on the first-page results. Examples are: query clarity $QClar_1$, query entropy $QEnt2$ and etc. Since we only have 10 documents on the first-page results, the calculation of language models based on both relevant and non-relevant documents can be computed efficiently for online prediction. Also, the query-based features can be computed efficiently for online prediction.

7.3.2 Learning Algorithm

We would like to use a learning technique to combine the features to predict the novelty parameter for diversification. In this dissertation, we use logistic regression [2] since it can take any input value and outputs a value between zero and one which is what we want. The logistic regression model is of the form: $f(z) = \frac{1}{1+\exp(-z)}$, where z is a set of features and $f(z)$ is the probability of an outcome. $f(z)$ is used to predict the novelty coefficient, i.e., λ. z is of the form $z = \bar{w}\bar{x}$ where $\bar{x}$ is a vector of the features and $\bar{w}$ is the learned weight (based on our training data) associated with each feature. When a weight is positive, it means that the corresponding feature increases the probability of outcome while a negative coefficient means the opposite. Once these weights are learned based on our training data (i.e., past queries and their optimal λ values as determined by some utility function to be defined in the next section), we can use the model (features along with their weights) to predict the diversity coefficient parameter (λ) for test queries. Statistical package R [3] is used to train the model.

[2] Since logistic regression has a global optimum, the choice of the learning algorithm is of little importance.
[3] http://www.r-project.org/

7.3.3 Evaluation Metric

Our goal is to optimize the utility to a user over all the three pages P_1, P_2, and P_3. We thus need to define how we measure the utility over all these pages. Such a measure is needed to evaluate the effectiveness of any method for optimizing exploration-exploitation tradeoff and also needed to set a criterion for selecting the optimal λ values for training in learning an adaptive diversification coefficient.

Since the first-page results is the same for all the methods, we only consider the utility on the second and third-page results. The user is assumed to see all the results on the second page, thus we define the utility on the second page as the number of relevant documents on the second page, denoted as $U(P_2) = REL(P_2)$.

For the third page, since the user is only assumed to see some results, the utility is defined as the expected number of relevant documents that the user would see on the third page:

$$U(P_3) = \sum_{j=1}^{10} s_j \sum_{i=1}^{j} \delta(i) \tag{7.16}$$

where s_j denotes the probability that the user would stop reading after seeing the top j documents and $\delta(i)$ is 1 if the i-th document is relevant and is zero otherwise.

If we assume the uniform stopping probability, i.e., s_j is uniform (i.e. $s_j = \frac{1}{10}$ since we have 10 documents per result page), we would model the impatient user [96] and the utility on the third page with some algebraic transformation is reduced to:

$$U(P_3) = \sum_{n=1}^{10} \frac{(11-n)}{10}.\delta(n) \tag{7.17}$$

where n is the ranking of the document. For example, if a document is ranked first on the third page, the expected utility for that document would be one since $n = 1$, similarly, if the document is ranked at 10-th, the expected utility from that document would be $\frac{1}{10}$.

This measure is reasonable for a user who is "impatient" who might indeed stop at any of the positions equally likely. However, if the user is more "patient" with a high probability of viewing all the results on the third page, we would like to put more weight on the utility from the third page. One way to account for that is to parameterize the weights on the third page so that we can examine this effect as follows:

$$U(P_3) = \sum_{n=1}^{10} [(\frac{(k-1)}{9}).(n-1).\delta(n) + 1] \tag{7.18}$$

where $k \in [0, 1]$. When $k = 0.1$, we gain Equation(7.17).

The total utility on both the second page and the third page is thus $U(P_2) + U(P_3)$.

7.4 Experiment Design

We used three standard TREC data sets in our study [4]: TREC2004 robust track, WT2G and AP88-90 which represent different genre of text collections. There were 249 queries [5] in TREC2004 robust track data set. WT2G is a web data set with 50 queries while AP88-90 is a homogeneous data set with 149 queries. We used Porter stemming and did stop word removal on all these collections.

Since we do not have a real system for interactive search, we simulated user-clicked documents (user feedback) as follows: We used Lemur toolkit to index document collections and retrieved a document list for each query using Kullback-Leibler divergence retrieval model with Dirichlet prior smoothing (in Section 3.1) to return 10 result documents (this is to simulate the first-page results). The reason for showing this page to the user is to get some limited information about relevant documents and extract features that we need for our learning algorithm. Also, the reason for returning only 10 documents for each page is to resemble the Web search engine that shows 10 results per page, so we also simulate to have 10 results per page. We then extract features from the first-page results to decide if we need to diversify the results on the second page and then feedback is used to improve the ranking of documents on the third page based on the user judgments on the first two pages. (We assume that the user would click on a relevant document.)

We compare our proposed methods (i.e., FixedDivFB and AdaptDivFB) with baseline methods, i.e., NoFeedback and RegularRelFB by measuring the total utility over all three pages. Since the first page is the same for all the methods, we actually only computed the utility over the second and third pages. While the first-page results are similar for all these four methods, the second-page results for different methods are different. For NoFeedback and RegularRelFB, we return 10 results using Kullback-Leibler divergence retrieval model [78]. For FixedDivFB, we use MixAvg-MMR [163] with a learned fixed novelty parameter (using training queries) for all test queries and for AdaptDivFB, we use adaptive novelty coefficient (by learning it with training data) for each individual query. Then for the third page, we use language model feedback [149, 164] to return 10 documents as the third-page results. Figure 7.1 shows these methods clearly.

To train our proposed methods, we need to obtain training data first. In our study, we used 5-fold cross validation to split our queries for testing and training purposes for each data set and get the average across all folds. Since we are restricted to vary the parameters in testing stage, we set the parameters to their optimal values gained from training data in the testing stage, i.e., we set the Dirichlet prior smoothing to its optimal value for each training data using NoFeedback method, and set the feedback coefficient to its optimal value using RegularRelFB. We fixed feedback term count to 100 and mixture noise parameter [164] to 0.9

[4] All experiments measured according to equation 7.17 unless otherwise stated.
[5] One query was dropped because the evaluators did not provide any relevant documents for it.

and β in negative feedback [149] to 0.1. We did not vary these parameters because this is not our purpose of study. For FixedDivFB, we choose an optimal fixed coefficient novelty, i.e., λ for all queries learned based on training data. We then use the optimal parameters gained from training data to measure the performance of test queries. The needed training data for AdaptDivFB is of the form of a set of feature vectors computed based on different queries and retrieval results along with the corresponding optimal novelty parameter. The learning task is to learn from such a training data set to predict the optimal novelty parameter for a new test query based on its corresponding feature values. To get the optimal novelty parameter for training queries for AdaptDivFB, we try different novelty coefficient $\lambda \in \{0, 0.1, 0.2, \ldots, 1\}$ using MixAvg-MMR [163] on training data sets and we choose the best λ for each training query to form our training data set. As a result, the main difference between the FixedDivFB and AdaptDivFB methods lies in what they can learn from the training data: the FixedDivFB learns a fixed novelty coefficient, λ, that leads to the best utility (described in Section 7.3.3) on the training data set, while the AdaptDivFB method learns a *prediction model* that best fits the training data set.

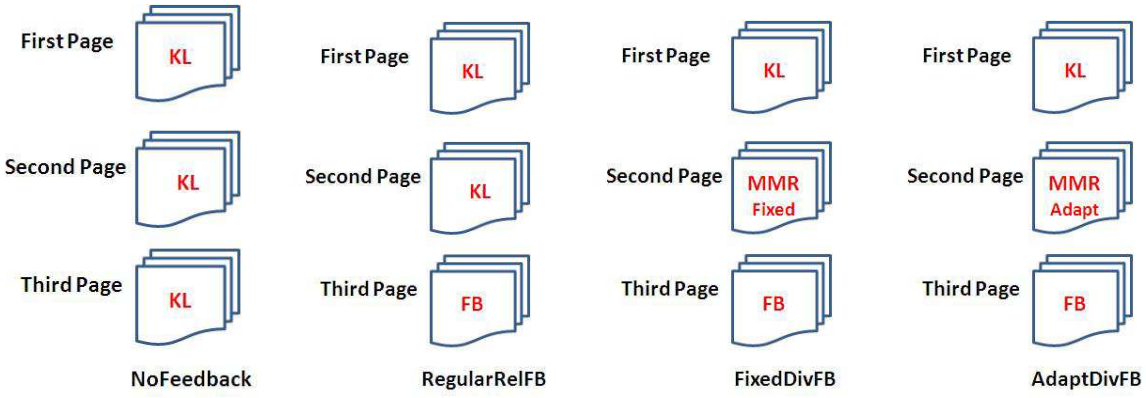

Figure 7.1: Visualization of different methods. KL means Kullback-Leibler divergence retrieval model [78], MMR Fixed means MixAvg-MMR using fixed novelty coefficient for all queries, MMR Adapt means MixAvg-MMR using adaptive novelty coefficient for each query and FB means language model feedback [149, 164].

7.5 Experimental Results

Our main hypothesis is that optimizing the exploration-exploitation tradeoff over a session is effective and we need to optimize such a tradeoff for each individual query. In order to test these hypotheses, we conduct a set of experiments. In section 7.5.1, we empirically examine the exploration-exploitation tradeoff. Next, in section 7.5.2, we examine the effectiveness of optimizing the total user utility over a session. In section 7.5.3, we go further to see the effect of optimizing either exploration or exploitation and compare the results when we optimize based on the combination. Then, in section 7.5.4, we examine what kind of queries would

benefit from optimizing the exploration-exploitation tradeoff. Next, we consider the effect of optimizing the exploration-exploitation and evaluate this effect on user patience. Since there are so many methods for diversification, in section 7.5.6 we consider a different diversification method and evaluate the effect of the diversification method on our main hypothesis. Finally, we analyze our features for optimizing the novelty coefficient.

7.5.1 Is Optimal Exploration-Exploitation Tradeoff Query Dependent?

Diversification methods are usually controlled by an interpolation novelty coefficient λ to control the balance between relevance and redundancy. According to the description in Section 7.1, when $\lambda = 1$, the emphasize is on *novelty* whereas for $\lambda = 0$, the emphasize is on *relevancy*. To show the sensitivity of λ to the utility function described on Section 7.3.3, we plot the utility of several randomly selected topics from each category, i.e., easy [6], difficult [7] and others (351, 421, 425) by varying λ from 0 to 1. The results are shown in Figure 7.2. These patterns show that the optimal exploration-exploitation tradeoff is query dependent and it is important to dynamically optimize the novelty coefficient in a per-query basis.

7.5.2 Effectiveness of Optimizing the Total User Utility over a Session

Our hypothesis is that in order to achieve optimal retrieval performance and outperform relevance feedback, we need to optimize based on the *total user utility*. In order to test our hypothesis, we compare our proposed methods (i.e., FixedDivFB and AdaptDivFB) with baselines (i.e., NoFeedback and RegularRelFB) which are optimized based on the whole utility, i.e., utilities on the second and third pages. Table 7.1 shows the comparison of our methods with baselines across different TREC data sets (we show the utilities based on the second, third and total (second+third) pages.). The results indicate that method RegularRelFB is better than NoFeedback as we expected since feedback outperforms the basic retrieval model using Kullback-Leibler divergence. Method FixedDivFB outperforms RegularRelFB and AdaptDivFB outperforms both FixedDivFB and RegularRelFB methods since it uses different novelty coefficient for each query. Statistical Significant tests using Wilcoxon singed-rank test indeed indicate that our proposed methods are statistically significant over method RegularRelFB [8]. The table also shows percentage improvement, e.g., FixedDivFB/RegularRelFB means the percentage improvement for FixedDivFB over RegularRelFB. Thus, by optimizing exploration-exploitation tradeoff, both FixedDivFB and AdaptDivFB outperform the regular

[6] We define a query as easy when its Precision@10 is 0.9 or 1, given a retrieval model.

[7] We define a query as difficult when its Precision@10 is no larger than 0.1, given a retrieval model.

[8] The reason why AdaptDivFB is not statistically significant over FixedDivFB for WT2G data set is because we only have 50 queries and since we do 5-fold cross validation, we have only 40 training queries in each fold which presumably is not sufficient for learning.

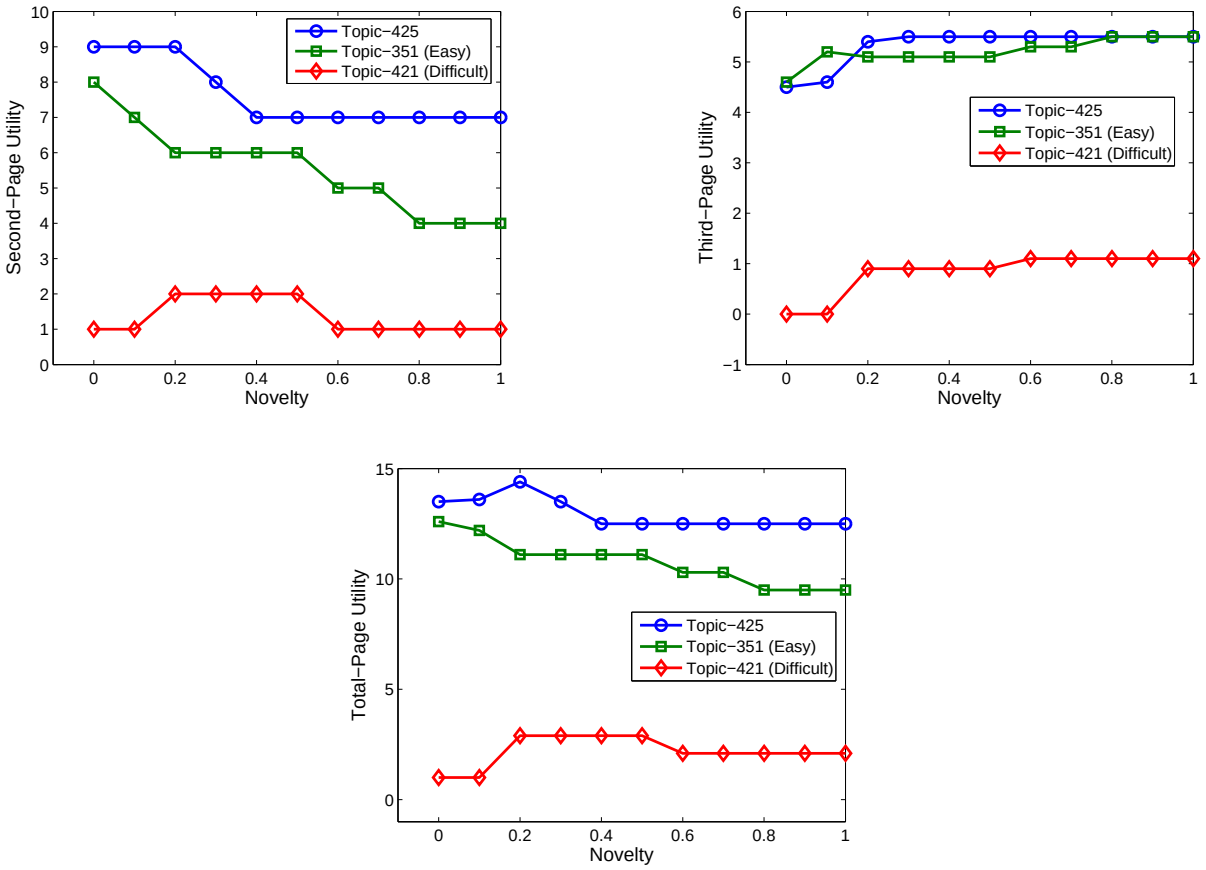

Figure 7.2: Exploration-exploitation tradeoff patterns. Optimal exploration-exploitation tradeoff is query dependent.

relevance feedback method, i.e., RegularRelFB which ignores exploration. And indeed AdaptDivFB outperforms FixedDivFB method which indicates that the optimal exploration-exploitation tradeoff is query dependent.

Please note that while it appears that AdaptDivFB is simply more effective than RegularRelFB even for the second page (where we would expect RegularRelFB to have some advantage), a decomposed analysis in Table 7.3 shows that the improvement on the second page comes from difficult queries, and AdaptDivFB does not perform as well as RegularRelFB for easy queries on the second page.

7.5.3 Detailed Analysis of Exploration-Exploitation Tradeoff

Our hypothesize is that to achieve optimal overall utility on a session, we should train with a "similar" objective function (i.e., a similar utility measure on the training data). Thus, we expect training to optimize utility on both second and third page should lead to better performance than training to optimize either one

Table 7.1: Comparison of different methods on different TREC data sets. $*$ and § mean significant over RegularRelFB and FixedDivFB, respectively.

	Robust 2004		
Methods	Second	Third	Total
NoFeedback	2.8593	1.2171	4.0764
RegularRelFB	2.8593	2.2749	5.1342
FixedDivFB	2.8513	2.3207	5.172*
FixedDivFB/RegularRelFB	-0.27%	2.01%	0.74%
AdaptDivFB	2.919	2.3511	**5.2701***§
AdaptDivFB/RegularRelFB	2.08%	3.34%	2.65%
AdaptDivFB/FixedDivFB	2.37%	1.31%	1.9%

	WT2G		
Methods	Second	Third	Total
NoFeedback	2.7	1.276	3.976
RegularRelFB	2.7	1.632	4.332
FixedDivFB	2.8	1.722	4.522*
FixedDivFB/RegularRelFB	3.7%	5.51%	4.385%
AdaptDivFB	2.84	1.7	**4.58***
AdaptDivFB/RegularRelFB	5.18%	6.61%	5.72%
AdaptDivFB/FixedDivFB	1.43%	1.045%	1.28%

	AP88-90		
Methods	Second	Third	Total
NoFeedback	2.5154	1.3149	3.8303
RegularRelFB	2.5154	2.4153	4.9307
FixedDivFB	2.558	2.4267	4.9847*
FixedDivFB/RegularRelFB	1.69%	0.47%	1.09%
AdaptDivFB	2.6633	2.4446	**5.1079***§
AdaptDivFB/RegularRelFB	5.88%	1.21%	3.59%
AdaptDivFB/FixedDivFB	4.12%	0.73%	2.47%

alone. In particular, optimizing the second-page "**only**" would lead to over-exploitation (ending up with lower third-page utility), while optimizing third-page "**only**" would be the opposite.

In order to see if this hypothesizes is true, we conduct two experiments: (1) optimizing the second-page utility **only** (exploitation) (2) optimizing the third-page utility **only** (exploration). The results are shown in Table 7.2 (we only show the results for AdaptDivFB method, similar patterns can also be seen for FixedDivFB method.). From the table, we see that across all data sets, the second-page utility once optimized on the second-page only is higher than when it is trained based on both pages. And indeed the third-page utility is lowered. The opposite trend could be explained when optimizing on the third-page only. The table also shows the percentage improvements, e.g., AdaptDivFB (2nd+3rd/2nd) indicates the percentage improvement when optimized on both pages vs. optimized on the second page only.

The other interesting observation is that the overall utility is degraded and this indeed indicates that

optimizing the total utility, i.e. both second and third-page utility, is necessary to lead to the *optimal* retrieval performance.

These observations indeed confirm our hypothesis that in order to have the optimal retrieval performance, the exploration-exploitation tradeoff needs to be optimized.

Table 7.2: Comparison of optimizing based on second-page utility, third-page utility and second+third utility for AdaptDivFB. $*$ and § mean significant over AdaptDivFB (2nd) and AdaptDivFB (3rd), respectively.

AdaptDivFB			
	Robust 2004		
Methods	Second	Third	Total
AdaptDivFB (2nd)	2.976	2.1905	5.1665
AdaptDivFB (3rd)	2.5249	2.4427	4.9676
AdaptDivFB (2nd+3rd)	2.919	2.3511	**5.2701***§
AdaptDivFB (2nd+3rd/2nd)	-1.91%	7.33%	2.005%
AdaptDivFB (2nd+3rd/3rd)	15.60%	-3.75%	6.09%

AdaptDivFB			
	WT2G		
Methods	Second	Third	Total
AdaptDivFB (2nd)	2.9	1.57	4.47
AdaptDivFB (3rd)	2.38	1.962	4.342
AdaptDivFB (2nd+3rd)	2.84	1.74	**4.58***§
AdaptDivFB (2nd+3rd/2nd)	-2.07%	10.82%	2.46%
AdaptDivFB (2nd+3rd/3rd)	19.32%	-11.31%	5.48%

AdaptDivFB			
	AP88-90		
Methods	Second	Third	Total
AdaptDivFB (2nd)	2.728	2.226	4.954
AdaptDivFB (3rd)	2.482	2.516	4.998
AdaptDivFB (2nd+3rd)	2.6633	2.4446	**5.1079***§
AdaptDivFB (2nd+3rd/2nd)	-2.37%	9.82%	3.11%
AdaptDivFB (2nd+3rd/3rd)	7.3%	-2.83%	2.19%

7.5.4 Best Queries for Tradeoff Optimization

Our hypothesis is that if a query is more *difficult*, it would benefit more from optimizing the exploration-exploitation tradeoff. Indeed, the results in Table 7.3 (these results are based on their counterpart results in table 7.1 for Robust 2004 data set.) confirm this hypothesis. We separate the results in Table 7.1 into easy and difficult queries and measure their performance.

As we see from these results, it is clear that optimizing the exploration-exploitation tradeoff helps difficult queries more than easy queries, i.e., both methods FixedDivFB and AdaptDivFB outperform baseline results for difficult queries but the improvement for easy queries is negligible.

Another observation from this table is that increasing diversity helps difficult queries due to the implied (desirable) negative feedback however, hurts easy queries because of the implied (incorrect) negative feedback.

Table 7.3: Comparison of methods for DIFFICULT and EASY queries on Robust 2004 Data Set, 51 difficult queries and 21 easy queries.$*$ and § mean significant over RegularRelFB and FixedDivFB, respectively.

DIFFICULT Queries			
Methods	Second	Third	Total
NoFeedback	0.7129	0.5208	1.2337
RegularRelFB	0.7129	1.4738	2.1867
FixedDivFB	0.8256	1.497	2.3226*
AdaptDivFB	0.9197	1.5091	**2.4288***§

EASY Queries			
Methods	Second	Third	Total
NoFeedback	6.7	2.6223	9.3223
RegularFB	6.7	3.901	10.601
FixedDivFB	6.45	4.02	10.47
AdaptDivFB	6.5333	4.068	**10.6013***

7.5.5 Exploration-Exploitation Tradeoff and User Patience

Since the standard relevance feedback does not do exploration, in "certain situations", i.e., when a user is more *patient*, exploration would have more benefit. When a user is more patient, there is a high probability of viewing all the results on the third page, as a result, putting more weight on the utility from the third page would be more beneficial for such a user. As we discussed in section 7.3.3, we model user patience with k (equation 7.18), so we now have a different utility measure which is parameterized with k. Our hypothesize is that for patient users, *exploration* is more useful.

In Figure 7.3, we vary k and measure the performance for RegularRelFB and FixedDivFB [9]. Figure 7.3 (left) shows the difference between these two methods; as k gets larger, the difference (i.e., FixedDivFB-RegularRelFB) between these methods is larger which indicates that the benefit from exploration is more amplified as k increases. Also, Figure 7.3 (right) shows their performance on the third page (to show the exploration benefit) for both methods, it also indicates that as k gets larger, the difference between these two methods is more amplified. (The linear trend curves are expected given the form of our utility function.)

[9] Similar trends can be seen for AdaptDivFB.

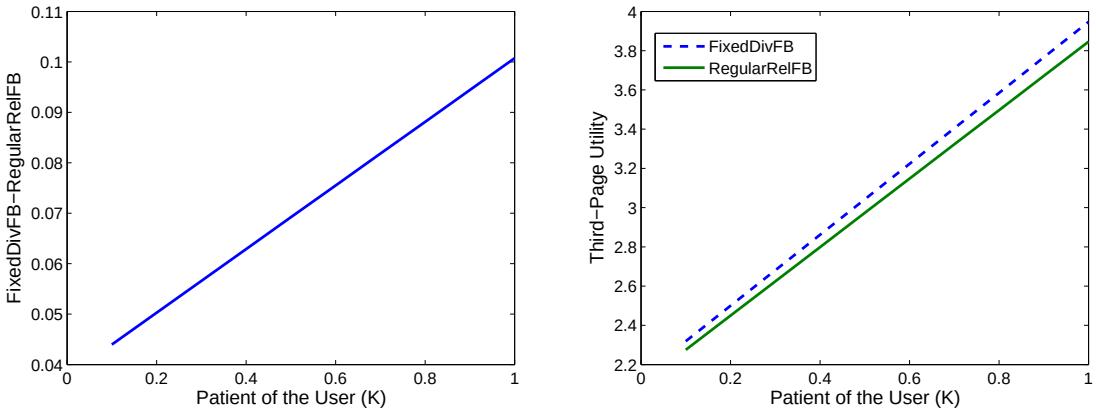

Figure 7.3: Modeling patience of the user (based on Robust 2004).

7.5.6 Sensitivity to Diversification Methods

We also want to know how our findings change if we use a different diversification method, i.e., MMR-PLSA proposed in [72]. In this method, both the documents and the query would be mapped to a low-dimensional space representation through Probabilistic Latent Semantic Indexing (PLSA) model [58], and then a similar greedy algorithm to Maximal Marginal Relevance (MMR) [16] is used to select a diverse set of documents. In this model, instead of indirectly covering multiple topic aspects through eliminating the redundancy among documents, an intuitively better strategy is to directly cover topic aspects of the document through explicitly modeling subtopics of the documents.

The results for this method are shown in Table 7.4. These results also support our *main finding*, i.e., optimizing *exploration-exploitation tradeoff* outperforms traditional relevance feedback, and adaptive optimization (AdaptDivFB) is better than fixed-coefficient optimization (FixedDivFB).

7.5.7 Analysis of Features for Optimizing Novelty Coefficient

In this section, we discuss about the feature distributions and how they are linked to the novelty coefficient. As discussed, we do 5-fold cross validation, and for each fold we have a different model according to our training data. So, in total, we have 15 models for method AdaptDivFB in Table 7.1 (3 data sets and 5-fold). For each model, we use AIC (Akaike Information Criterion) [53] to include only statistical significant features. In order to understand the correlations of features to the novelty parameter, in Table 7.5, we show the distributions across all 15 models. Negative means there is a negative correlation between that feature and novelty parameter whereas positive means the opposite. As shown in the table, features $QEnt1$, $QClar2$, $VirAP$ and Div are the most frequently used features by these 15 models.

Table 7.4: Comparing different methods and using MMR-PLSA as a diversification method. $*$ and § mean significant over RegularRelFB and FixedDivFB, respectively.

Robust 2004			
Methods	Second	Third	Total
NoFeedback	2.8593	1.2117	4.0764
RegularRelFB	2.8593	2.2749	5.1342
FixedDivFB	2.8511	2.3597	5.2108*
AdaptDivFB	2.911	2.4007	**5.3117***§

WT2G			
Methods	Second	Third	Total
NoFeedback	2.7	1.276	3.976
RegularRelFB	2.7	1.632	4.332
FixedDivFB	2.86	1.732	4.92*
AdaptDivFB	3	1.776	**4.776***§

AP88-90			
Methods	Second	Third	Total
NoFeedback	2.5154	1.3149	3.8303
RegularRelFB	2.5154	2.4153	4.9307
FixedDivFB	2.604	2.401	5.005*
AdaptDivFB	2.6377	2.4059	**5.0436***

The coefficients in this table are consistent with what we discussed in section 7.3.1, i.e., if a query is long ($|Q|$), or the similarity between relevant and non-relevant documents is high (CosSim), or when the results are not sufficiently diversified (Div), we need diversification. In other cases, i.e., when a query is clear enough (QClar2) or relevant documents contain broad topics (QEnt1), we do not need diversification. The contradictory effect of negative coefficient of $QClar1$ and positive coefficient of $exp(QClar1)$ could be explained as follows: in most models these two features co-occur with each other and the coefficients are such that when a query is *clear* enough, i.e., $QClar1$ is relatively large, the overall effect would be negative suggesting that we should use a smaller novelty coefficient. However, when a query is not clear enough, the overall effect is positive which suggests we need to use a larger diversity coefficient. The interesting observation of negative influence of VirAP indeed confirms our hypothesis that for difficult queries, we need more diversification.

7.6 Chapter Summary

In this chapter, we studied how to optimize relevance feedback to maximize the total utility over an entire interaction session. In particular, we studied the issue of exploration-exploitation tradeoff in interactive

Table 7.5: Feature distributions

	Negative	Positive
$\|Q\|$	-	20%
$QDist$	33.3%	-
$QEnt_1$	53.3%	-
$QEnt_2$	26.6%	-
$QClar_1$	20%	-
$QClar_2$	40%	-
num	33.3%	-
JSD	26.6%	-
CosSim	-	33.3%
Div	-	40%
VirAP	53.3%	-
$\exp(QClar_1)$	-	26.6%

feedback, which is the tradeoff between presenting search results with the highest immediate utility to a user and presenting search results with the best potentials for collecting feedback information. We framed this tradeoff as a problem of optimizing the diversification of search results. We proposed two methods that optimize the exploration-exploitation tradeoff. The first method is to fix a novelty coefficient for diversification and the other one is to adaptively optimizing the diversification of search results for each query. We also defined utility from user perspective and defined how we can model both patient and impatient users. Experimental results on three representative TREC data sets indicate that our proposed methods are effective for optimizing the tradeoff between exploration and exploitation and outperform the traditional relevance feedback which only does exploitation without exploration. In summary, our findings are as follows:

- Optimal exploration-exploitation tradeoff is query dependent and it is important to dynamically optimize the novelty coefficient in a per-query basis.e
- In order to achieve optimal retrieval performance and outperform relevance feedback, we need to optimize based on the total user utility. In other words, in order to have the optimal retrieval performance, the exploration-exploitation tradeoff needs to be optimized.
- When a user is more patient, exploration would have more benefit because there is a high probability of viewing all the results on later pages.
- If a query is more difficult, it would benefit more from optimizing the exploration-exploitation tradeoff.

One limitation of our study is that most users of a current Web search engine do not view so many pages, even though in the case of a high-recall search task or when a user uses a small-screen device (e.g., a smartphone), we can expect a user to often view more than two pages of results. Thus it would be interesting

to consider more realistic assumptions such as considering fewer results per page to simulate the smartphone scenario. Also, it would be interesting to evaluate the methods using the actual click-through data from query logs. Ideally, building a real system that involves real users in interaction would be interesting.

Another limitation of our work is that in our current formulation, we assumed the availability of some limited feedback information from the first-page result to make the problem more tractable; an interesting future direction would be to just get information about the query which would also increase the applicability of our method, but the problem would be that the only information we have available is information regarding the query and we do not yet know anything about the relevant/non-relevant documents, so it would also be harder to solve the problem of optimizing the diversity parameter.

In the current problem formulation, we only consider the second-page when measuring the novelty and we only use the feedback information from the second page to re-rank results on the third page, so the diversity on the second page is what matters in terms of optimizing the exploration-exploitation tradeoff. For future, it is interesting to explore optimizing the diversity in the combined set of first and second pages.

In optimizing the exploration-exploitation tradeoff, the generated diversified results are based on the greedy MMR approach which models novelty in respect to the previously selected documents, i.e., it indirectly optimizes diversity by removing redundancy. An alternative way is to directly maximize the coverage of different aspects of a topic. For example, we can formulate the task of diversified retrieval as the problem of predicting the diverse subsets. Specifically, we can formulate a discriminant based on maximizing word coverage, and perform training using the learning to rank methods.

Chapter 8

Summary and Future Work

This dissertation studied how to systematically optimize the accuracy of Web search engines for difficult queries. Due to the lack of necessary domain knowledge or complex information needs, a user can often encounter difficulty to compose effective queries. We proposed novel algorithms to systematically improve the retrieval accuracy in three important directions naturally corresponding to different stages of an interactive search process:

- **Pre-retrieval**: Where we proposed novel and efficient estimation methods to bridge the vocabulary gap when the user information need is not expressed exactly as it is in the relevant documents in statistical translation language model framework. We then analyzed the model more to gain insight about the optimality of the translation language model by introducing constraints that any reasonable estimation method should satisfy. We also studied an application of statistical translation language model for Twitter search which further confirms that statistical translation language model can bridge the vocabulary gap and it can also improve the estimate of term frequency.

- **Post-retrieval**: After we present the results to the users, gaining feedback from the users would help improve the retrieval accuracy especially for difficult queries. In a case when a query is very difficult that the search results are poor that none of the top-ranked documents are relevant, we proposed an optimization framework in which we learn from a few top-ranked negative documents and search in the space of all candidate language models to build a more general negative language model to prune aggressively but carefully a lot of non-relevant documents from the top-ranked documents in a language modeling framework.

- **Interactive retrieval**: When a query is difficult, the user is unlikely satisfied with only one interaction with the search engine; so there will be more interactions to get the desired results. As a result, after we present the results to the user (i.e., post-retrieval stage), there is a tradeoff between presenting search results with the highest immediate utility to a user (but not necessarily most useful for collecting feedback information) and presenting search results with the best potential for collecting useful feedback

information (but not necessarily the most useful documents from a user's perspective). We proposed methods to optimize such a tradeoff (exploration-exploitation tradeoff) of the overall utility of feedback to a user in the entire session of user interaction.

We now discuss how related research fields can also benefit from the methods proposed in this dissertation, as well as more general information retrieval problems:

Formalize more translation language model constraints: Ideally, if we could enumerate all desirable translation language model constraints, the target translation language model function would be guaranteed to be optimal. Unfortunately, it is extremely difficult, if not impossible, to find all desirable constraints. However, even if we could not find the complete set of desirable constraints, such incomplete set of constraints are still useful for us to find a more robust and effective translation language model.

Query adaptive retrieval models: Many studies in information retrieval show that no single retrieval model is able to return satisfactory results for every query. The main reason is that the existing retrieval models fail to adjust their scoring functions dynamically based on the queries. It would thus be interesting to study how to automatically categorize queries based on the causes and automatically adapt the strategy to improve search results for difficult queries based on identified causes. The current understanding of the causes of difficult queries are still limited in a coarse granularity. In future, constructing a comprehensive and finer-granularity taxonomy of difficult queries would have high impact. Based on the taxonomy, machine learning techniques can be leveraged to identify or categorize difficult queries into the taxonomy which we can further study specific retrieval models to improve search utilities for a family of difficult queries.

Unified search paradigm: In this dissertation, we studied how to optimize the accuracy of Web search engines for difficult queries from naturally three different perspectives which we showed their effectiveness individually. The proposed methods for improving a search engine in pre-retrieval, post-retrieval, and interactive search are orthogonal, thus they can be combined in a search engine to potentially achieve additive benefit. However, how to integrate all these approaches in a unified formal framework remains a major open challenge. This direction involves interesting research questions such as, human-computer interface (HCI) design, decision making, redundancy control, etc. Such a system makes it possible to collect massive user feedback information which enables us to build a social system where past user interactions can benefit future information seekers.

One clear advantage of such a system is the potential benefit from improved estimation of the models based on additional training data. As search engines are being used, we will be able to collect a lot of implicit feedback information such as click-throughs. Translation language models appear especially to be promising in this direction and they are complementary with the discriminative models for learning to rank models. In addition, queries and the associated clicked documents can naturally serve as training data to train a translation language model in a supervised way.

Novel applications: As our society becomes more data-driven, applications and tasks of all types will come to increasingly rely upon information systems. Information services can potentially aid us in every aspect of our lives. But our growing number of systems and services are becoming ever more difficult to maintain. This line of research can lead to cost-effective information systems that can efficiently adapt to variety of retrieval domains such as enterprise search, library search, medical search, and the many new and exciting applications to come.

References

[1] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. *In Proceedings of Neural Information Processing Systems*, pages 1–8, 2006.

[2] D. Agarwal, B. Chen, and P. Elango. Explore/exploit schemes for web content optimization. *In Proceedings of IEEE International Conference on Data Mining*, 2009.

[3] R. Agrawal, S. Gollapudi, A. Halverson, and S. Leong. Diversifying search results. *In Proceedings of ACM International Conference on Web Search and Data Mining*, pages 5–14, 2009.

[4] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *Journal of ACM Transactions on Information Systems*, pages 20:357–389, 2002.

[5] R. Attar and A. S. Fraenkel. Local feedback in full-text retrieval systems. *Journal of the ACM*, pages 24(3):397–417, 1977.

[6] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 44–54, 2006.

[7] J. Bai, D. Song, P. Bruza, J. Y. Nie, and G. Cao. Query expansion using term relationships in language models for information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 688–695, 2005.

[8] A. Berger and J. Lafferty. Information retrieval as statistical translation. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229, 1999.

[9] A. Bookstein. Information retrieval: A sequential learning process. *Journal of American Society (ASIS)*, pages 34(5):331–342, 1983.

[10] P. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Journal of Computational Linguistics*, 19(2):263–311, 1993.

[11] C. Buckeley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using smart: Trec3. *In Proceedings of Text REtrieval Conference*, pages 69–80, 1994.

[12] C. Buckley. Why current ir engines fail. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 584–585, 2004.

[13] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. *In Proceedings of International Conference on Machine Learning*, pages 89–96, 2005.

[14] G. Cao, J. Y. Nie, and J. Bai. Integrating word relationships into language models. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–305, 2005.

[15] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. *In Proceedings of International Conference on Machine Learning*, pages 129–136, 2007.

[16] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[17] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 390–397, 2006.

[18] B. Carterette and P. Chandar. Probabilistic models of novel document rankings for faceted topic retrieval. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 1287–1296, 2009.

[19] B. Carterette and D. Petkova. Learning a ranking from pairwise preferences. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 629–630, 2006.

[20] H. Chen and D. R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 429–436, 2006.

[21] C. L.A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Bttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, 2008.

[22] P. Clough, M. Sanderson, M. Abouammoh, S. Navarro, and M. L. Paramita. Multiple approaches to analysing query diversity. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 734–735, 2009.

[23] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, pages 4:129–145, 1996.

[24] K. Collins-Thompson. Estimating robust query models using convex optimization. *In Proceedings of Neural Information Processing Systems*, 2008.

[25] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 837–846, 2009.

[26] K. Collins-Thompson and P. N. Bennett. Estimating query performance using class predictions. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 672–673, 2009.

[27] K. Collins-Thompson and J. Callan. Query expansion using random walk models. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 704–711, 2005.

[28] W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic retrieval based on staged logistic regression. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 198–210, 1992.

[29] T. Cover and J. Thomas. Elements of information theory. *John Wiley and Sons, New York, USA*, 1991.

[30] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, 2002.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 39(B):1–38, 1997.

[32] J. V. Dillon and K. Collins-Thompson. A unified optimization framework for robust pseudo-relevance feedback algorithms. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 1069–1078, 2010.

[33] Ecology. http://webecologyproject.org/.

[34] M. Efron. Hashtag retrieval in a microblogging environment. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 787–788, 2010.

[35] M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 495–504, 2011.

[36] T. Ellman. Explanation-based learning: a survey of programs and perspectives. *ACM Computing Surveys*, 21:163–221, June 1989.

[37] V. Faber. Clustering and the continuous k-means algorithm. *Los Alamos Science*, pages 138–144, 1994.

[38] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–56, 2004.

[39] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *Journal of ACM Transactions on Information Systems*, 29, 2011.

[40] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 480–487, 2005.

[41] H. Fang and C. Zhai. Semantic term matching in axiomatic approaches to information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–122, 2006.

[42] N. Fuhr. Language models and uncertain inference in information retrieval. *In Proceedings of the Language Modeling and IR workshop.*

[43] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[44] N. Fuhr. A probability ranking principle for interactive information retrieval. *Information Retrieval Journal*, pages 11(3):251–265, 2008.

[45] N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *Journal of ACM Transactions on Information Systems*, pages 9(3):223–248, 1991.

[46] R. Fung and B. D. Favero. Applying bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–48, 1995.

[47] F. C. Gey. Inferring probability of relevance using the method of logistic regression. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–231, 1994.

[48] W. Goffman. A searching procedure for information retrieval. *Journal of Information Storage and Retrieval*, pages 2:73–78, 1964.

[49] S. Gouws, D. Metzler, C. Cai, and E. Hovy. Contextual bearing on linguistic variation in social media. *Workshop on Languages in Social Media*, pages 20–29, 2011.

[50] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *In Proceedings of International World Wide Web Conference*, pages 491–501, 2004.

[51] D. Harman. Relevance feedback revisited. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–10, 1992.

[52] D. Harman and C. Buckley. Sigir 2004 workshop: Ria and where can ir go from here. *SIGIR Forum*, pages 38(2):45–49, 2004.

[53] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, USA, 2001.

[54] C. Hauff, V. Murdock, and R. Baeza-Yates. Improved query difficulty prediction for the web. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 439–448, 2008.

[55] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. *In Proceedings of Symposium on String Processing and Information Retrieval*, pages 43–54, 2004.

[56] J. He, M. Larson, and M. De Rijke. Using coherence-based measures to predict query difficulty. *In Proceedings of European Conference on Information Retrieval*, pages 689–694, 2008.

[57] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[58] T. Hofmann. Probabilistic latent sematic indexing. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[59] T. Jaakkola and H. Siegelmann. Active information retrieval. *In Proceedings of Neural Information Processing Systems*, 2001.

[60] F. Jelinek. *Statistical Methods for speech recognition*. MIT Press., 1997.

[61] E. C. Jensen, S. M. Beitzel, D. Grossman, O. Frieder, and A. Chowdhury. Predicting query difficulty on the web by learning visual clues. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 615–616, 2005.

[62] R. Jin, A. Hauptmann, and C. Zhai. Title language model for information retrieval. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–48, 2002.

[63] Y. Jing and B. Croft. An association thesaurus for information retrieval. *In Proceedings of RIAO Conference Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 141–160, 1994.

[64] T. Joachims. Optimizing search engines using clickthrough data. *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[65] T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. *In Proceedings of International Joint Conference on Artificial Intelligence*, 1997.

[66] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1 and part 2. *Information Processing and Management*, pages 36(6):779–808 and 809–840, 2000.

[67] M. Karimzadehgan and C. Zhai. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 323–330, 2010.

[68] M. Karimzadehgan and C. Zhai. Exploration-exploitation tradeoff in interactive relevance feedback. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 1397–1400, 2010.

[69] M. Karimzadehgan and C. Zhai. Improving retrieval accuracy of difficult queries through generalizing negative document language models. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 27–36, 2011.

[70] M. Karimzadehgan and C. Zhai. Axiomatic analysis of translation language model for information retrieval. *In Proceedings of European Conference on Information Retrieval*, pages 268–280, 2012.

[71] M. Karimzadehgan and C. Zhai. A learning approach to exploration-exploitation tradeoff in interactive relevance feedback. *Information Retrieval Journal*, 2012.

[72] M. Karimzadehgan, C. Zhai, and G. Belford. Multi-aspect expertise matching for review assignment. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 1113–1122, 2008.

[73] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[74] J. Kleinberg. The small-world phenomenon: an algorithm perspective. *In Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170, 2000.

[75] P. R. Kumar and P. P. Varaiya. Stochastic systems: Estimation, identification, and adaptive control. *Prentice Hall*, 1986.

[76] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 194–201, 2004.

[77] K. L. Kwok. A network approach to probabilistic information retrieval. *ACM Transactions on Office Information System*, 13:324–353, 1995.

[78] J. Lafferty and C. Zhai. Document language models, query models and risk minimization for information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, 2001.

[79] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. *In W. B. Croft and J. Lafferty, editors, Language Modeling and Information Retrieval. Kluwer Academic Publishers*, 2003.

[80] V. Lavrenko, M. Choquette, and B. Croft. Cross-lingual relevance models. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–182, 2002.

[81] V. Lavrenko and B. Croft. Relevance-based language models. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127, 2001.

[82] M. Lesk and B. Croft. Word-word associations in document retrieval systems. *American Documentation*, 20:20–27, 1969.

[83] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 177–187, 2005.

[84] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. *In proceedings of SIAM International Conference on Data Mining*, 2007.

[85] A. Leuski. Interactive information organization: Techniques and evaluation. *PhD Thesis, University of Massachusetts*, 2001.

[86] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[87] X. Lin and B. Croft. Cluster-based retrieval using language models. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, 2004.

[88] S. Liu, F. Lin, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 266–272, 2004.

[89] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, pages 1:309–317, 1957.

[90] Y. Lv and C. Zhai. Adaptive relevance feedback in information retrieval. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 255–264, 2009.

[91] R. Mandala, T. tokunaga, H. Tanaka, and K. Satoh. Ad hoc retrieval experiments using wordnet and automatically constructed thesauri. *In Proceedings of Text REtrieval Conference*, pages 475–481, 1998.

[92] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, pages 49(4):41–46, 2006.

[93] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. *In Proceedings of International Conference on Machine Learning*, pages 350–358, 1998.

[94] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. *In Proceedings of International World Wide Web Conference*, pages 101–110, 2008.

[95] D. Metzler, Y. Bernstein, B. Croft, A. Moffat, and J. Zobel. Similarity measures for tracking information flow. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 517–524, 2005.

[96] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *Journal of ACM Transactions on Information Systems*, page 27(1), 2008.

[97] V. Murdock and B. Croft. Simple translation models for sentence retrieval in factoid question answering. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 31–35, 2004.

[98] J. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–81, 1999.

[99] S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. *In Proceedings of International Conference on Machine Learning*, pages 721–728, 2007.

[100] S. Pandey, S. Roy, C. Olston, J. Cho, and S. Chakrabarti. Shuffling a stacked deck: The case for partially randomized ranking of search engine results. *In proceedings of Very Large Data Bases*, pages 781–792, 2005.

[101] M. L. Paramita, M. Sanderson, and P. Clough. Diversity in photo retrieval: Overview of the imageclefphoto task 2009. *In Proceedings of the international conference on Cross-language evaluation forum: multimedia experiments*, 2009.

[102] H. J. Peat and P. Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *J. of Information science*, 42(5):378–383, 1991.

[103] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[104] M. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, 14(3), 1980.

[105] Y. Qiu and H. Frei. Concept based query expansion. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169, 1993.

[106] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 691–692, 2006.

[107] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 239–248, 2005.

[108] F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. *In Proceedings of International Conference on Machine Learning*, pages 784–791, 2008.

[109] K. Raman, R. Udupa, P. Bhattacharya, and A. Bhole. On improving pseudo-relevance feedback using pseudo-irrelevant documents. *In Proceedings of European Conference on Information Retrieval*, pages 573–576, 2010.

[110] B. Ribeiro and R. Muntz. A belief network model for ir. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, 1996.

[111] B. Ribeiro-Neto, I. Silva, and R. Muntz. Bayesian network models for information retrieval. *Soft Computing in Information Retrieval: Techniques and Applications,*, pages 259–291, 2000.

[112] C. J. V. Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6), 1986.

[113] C. J. Van Rijsbergen. Information retrieval. *Butterworths*, 1979.

[114] S. E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, pages 33(4):294–304, 1977.

[115] S. E. Robertson and K. Sparck. Relevance weighting of search terms. *Journal of American Society for Information Science*, 27:129–146, 1976.

[116] S. E. Robertson and S. Walker. Some simple effective approximation to the 2-poisson model for probabilistic weighted retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, 1994.

[117] S. E. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 42–49, 2004.

[118] J. Rocchio. Relevance feedback in information retrieval. *In the SMART Retrieval System*, pages 313–323, 1971.

[119] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. *In Proceedings of International Conference on Machine Learning*, pages 441–448, 2001.

[120] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[121] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of Information Science*, pages 41(4):288–297, 1990.

[122] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill., 1983.

[123] G. Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *Journal of American Society for Information Science*, 26(1):33–44, 1975.

[124] M. Sanderson, J. Tang, T. Arni, and P. Clough. What else is there? search diversity examined. *In Proceedings of European Conference on Information Retrieval*, pages 562–569, 2009.

[125] A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy. Ranking mechanisms in twitter-like forums. *In Proceedings of ACM International Conference on Web Search and Data Mining*, pages 21–30, 2010.

[126] J. Savoy. Why do successful search systems fail for some topics. *In Proceedings of the 2007 ACM symposium on Applied computing*, pages 872–877, 2007.

[127] F. Scholer and S. Garcia. A case for improved evaluation of query difficulty prediction. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 640–641, 2009.

[128] H. Schutze and J. O. Pedersen. A co-occurrence based thesaurus and two applications to information retrieval. *Information and processing management*, 33(3):307–318, 1997.

[129] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50, 2005.

[130] X. Shen and C. Zhai. Active feedback in ad hoc information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–66, 2005.

[131] S. Singh, D. Litman, M. Kearns, and M. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, pages 105–133, 2002.

[132] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.

[133] A. Singhal, M. Mitra, and C. Buckley. Learning routing queries in a query zone. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32, 1997.

[134] A. F. Smeaton and C. J. Van Rijsbergen. The retrieval effects of query expansion on a feedback document retrieval system. *The Computer Journal*, 26(3):239–246, 1983.

[135] Software. https://github.com/lintool/twitter-corpus-tools.

[136] R. Song, Z. Luo, J.-R. Wen, Y. Yu, and H.-W. Hon. Identifying ambiguous queries in web search. *In Proceedings of International World Wide Web Conference*, pages 1169–1170, 2007.

[137] J. Tang and M. Sanderson. Evaluation and user preference study on spatial diversity. *In Proceedings of European Conference on Information Retrieval*, pages 179–190, 2010.

[138] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162–169, 2006.

[139] S. B. Thrun. The role of exploration in learning control. *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, 1992.

[140] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *In Proceedings of International Conference on Machine Learning*, pages 999–1006, 2000.

[141] Tunkrank. http://tunkrank.com/.

[142] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[143] Twitter. https://twitter.com/.

[144] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer Yahia. Efficient computation of diverse query results. *In Proceedings of IEEE International Conference on Data Engineering*, pages 228–236, 2008.

[145] E. M. Voorhees. Overview of the trec 2004 robust retrieval track. *In Proceedings of Text REtrieval Conference*, 2004.

[146] E. M. Voorhees. Draft: Overview of the trec 2005 robust retrieval track. *In Proceedings of Text REtrieval Conference*, 2005.

[147] E. M. Voorhess. Query expansion using lexical-semantic relations. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, 1994.

[148] X. Wang, H. Fang, and C. Zhai. Improve retrieval accuracy for difficult queries using negative feedback. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 991–994, 2007.

[149] X. Wang, H. Fang, and C. Zhai. A study of methods for negative relevance feedback. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 219–226, 2008.

[150] X. Wei and B. Croft. Lda-based document models for ad-hoc retrieval. In *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185, 2006.

[151] M. J. Welch, U. Schonfeld, D. He, and J. Cho. Topical semantics of twitter links. *In Proceedings of ACM International Conference on Web Search and Data Mining*, pages 327–336, 2011.

[152] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. *In Proceedings of ACM International Conference on Web Search and Data Mining*, 2010.

[153] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[154] S. K. M. Wong and Y. Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems,*, 13(1):69–99, 1995.

[155] J. Xu and B. Croft. Query expansion using local and global document analysis. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

[156] J. Xu, R. Weischedel, and C. Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 105–110, 2001.

[157] Z. Xu and R. Akella. A bayesian logistic regression model for active relevance feedback. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 227–234, 2008.

[158] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. *In Proceedings of European Conference on Information Retrieval*, pages 246–257, 2007.

[159] X. Xue, J. Jeon, and B. Croft. Retrieval models for question and answer archives. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–482, 2008.

[160] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 512–519, 2005.

[161] Y. Yue and T. Joachims. Predicting diverse subsets using structural svms. *In Proceedings of International Conference on Machine Learning*, pages 1224–1231, 2008.

[162] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. *In Proceedings of International Conference on Machine Learning*, pages 1201–1208, 2009.

[163] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17, 2003.

[164] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. *In Proceedings of ACM International Conference on Information and Knowledge Management*, pages 403–410, 2001.

[165] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, 2001.

[166] W. Zhang and T. Dietterich. A reinforcement learning approach to job-shop scheduling. *In Proceedings of International Joint Conference on Artificial Intelligence*, pages 1114–1120, 1995.

[167] Y. Zhang, W. Xu, and J. Callan. Exploration and exploitation in adaptive filtering based on bayesian active learning. *In Proceedings of International Conference on Machine Learning*, pages 896–903, 2003.

[168] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. *In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 287–294, 2007.

[169] X. Zhu, Andrew B. Goldberg, J. Van Gael, and D. Andrzejewski. Improving diversity in ranking using absorbing random walks. *In Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 97–104, 2007.

[170] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. *In Proceedings of International World Wide Web Conference*, pages 22–32, 2005.