EXPLOITING CONTEXT FOR LONG-TERM INFORMATION RETRIEVAL RELATED TASKS

by

Peilin Yang

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Engineering

Fall 2016

EXPLOITING CONTEXT FOR LONG-TERM INFORMATION RETRIEVAL RELATED TASKS

by

Peilin Yang

Approved: ______________________________
Xxxx Xxxx, Highest Degree
Chair of the Department of Xxxx

Approved: ______________________________
Xxxx Xxxx, Highest Degree
Dean of the College of Xxxx

Approved: ______________________________
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Xxxx Xxxx, Highest Degree
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Xxxx Xxxx, Highest Degree
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Xxxx Xxxx, Highest Degree
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Xxxx Xxxx, Highest Degree
Member of dissertation committee

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES vii
LIST OF FIGURES viii
ABSTRACT ix

Chapter

1 INTRODUCTION 1

1.1 Unified IR Evaluation System 2
1.2 Boundary Theory of Bag-of-Terms Models 3
1.3 Contextual Suggestion 4
1.4 Summary 6

2 RELATED WORK 7

2.1 Unified IR Evaluation System 7
2.2 Boundary Theory of Bag-of-Terms Models 9
2.3 Contextual Suggestion 10

2.3.1 Recommendation Systems 11
2.3.2 Text Summarization 13

3 REPRODUCIBILITY STUDY USING UNIFIED IR EVALUATION SYSTEM 14

4 BOUNDARY THEORY OF IR RANKING MODELS 15

5 CONTEXTUAL SUGGESTION 16

5.1 Introduction 17
5.2 Problem Formulation 17
5.3 Method with Category and Description 19

5.3.1 Gathering Candidate Suggestions 19

5.3.2 Ranking based on User Profiles 19

5.3.2.1 Category-based similarity: 20
5.3.2.2 Description-based similarity: 21

5.3.3 Ranking the candidates based on the contexts 21

6 FUTURE WORK **23**

BIBLIOGRAPHY **24**

LIST OF TABLES

5.1 **Examples of Categories in Example Suggestions** 20

LIST OF FIGURES

ABSTRACT

Information Retrieval (IR) is one of the most evolving research fields and has drawn extensive attention in recent years. Typically, there are different types of IR systems targeting to meet different users' information needs. We are more interested in two kinds of IR systems: (1) the system for IR researchers who are interested in understanding the IR ranking models, (2) the contextual suggestion system mainly for end users. In order to make better aforementioned IR systems that are satisfied for both researchers and end users, in our work we explorer the impact of context in two specific long-term IR related tasks – ranking model performance upper bound analysis using unified evaluation system and contextual suggestion.

A long standing problem of IR community is that newly proposed ranking models are implemented and evaluated using various systems of researchers' choices and such choices lead to the fact that different researchers reported different results for the same baseline model [64]. The context – IR systems used by researchers, is apparently the cause of such problem. This fact casts doubts on the real effectiveness of the proposed ranking models and we argue that it is essential for the IR community to have a unified evaluation system so that ranking models can be compared without considering the details – usually other components of an IR system, e.g. preprocessing of documents, choice of evaluation metrics. In our work we propose two unified IR evaluation systems to alleviate this problem. VIRLab as our first attempt. It provides an web based IR evaluation system where user of the system can implement her/his our model and get it automatically evaluated. Another system RISE (Reproducible IR System Evaluation) is proposed later to reflect the other endeavor from us. RISE is the first ever known system that users of the system can collaborate with each other and thus make the normalization of baseline models much more easier and thus can be

trusted by future researchers. A large scale reproducibility experiment based on RISE serves as the the performance reference of most widely used IR models.

Based on the reproducibility experiment results of using RISE we find that the optimum performances of several TREC collections do not improve a lot for over 20 years if bag-of-terms ranking model is used. This introduces another interesting question: does it exist the performance upper bound for bag-of-terms models? To answer this question we extensively investigate the context – the bag-of-terms document representation assumption and the statistics like term frequency and inverted document frequency the models usually play with. We first find that for single term queries several ranking models can be transformed to a simplified model. We then apply the cost/gain analysis which is commonly used in learning-to-rank (LTR) domain to find the optimum of single term queries for this simplified model. The result shows that although the performances of state-of-the-art ranking models are quit close to the practical optimum there is still some room for improvement.

For contextual suggestion system it is commonly agreed that how to model user profile is the key to the solution. We argue that user's preference which is usually modeled as a static or long term factor should also be considered as part of the context of this problem. Our first approach is to model the user profile using the venue's category and description from user's activity history. We further improve the method by leveraging the opinions from user's activity history to model the user profile. Such methodology utilizes the rich text associated with the users which naturally fits the problem into IR domain. By modeling the candidate suggestions in the similar fashion, the similarities between the candidates and the user profile are used to score the candidates. Experiments on TREC collections and a Yelp data collection confirm the advantage of our method.

For the future work, there are mainly two directions we would like to explore more. The first one is to quantify the impact of the context of unified IR evaluation system. There is no previous work on how much difference does the usage of different retrieval tools bring. We hope to be the first to report on standardizing and quantifying the impact so that the IR community could be aware of such divergence and can better evaluate the contributions of using various tools. The second one is to provide more sound justification about the boundary theory of ranking model performance. Specifically, we want to extend the current analysis on the single term queries to multiple term queries. We also would like to try other method, e.g. explanatory analysis, to achieve the same goal.

Chapter 1

INTRODUCTION

The past decades have witnessed the tremendous success of World Wide Web. People all over the world can now access to publicly available information via commercial search engines such as Google or Baidu with great ease. According to the online statistics[1], Google now (as of October 2016) can handle over 40,000 search queries every second on average, which translates to over 3.5 billion searches per day and 1.2 trillion searches per year worldwide. With such huge volume of search activities it is essential to make the search results of high quality in order to meet the users needs.

Information Retrieval (IR), usually used by academia in favor of its industrial counterpart search engine, is one of the most evolving fields and has drawn extensive attention in recent years. General speaking, there are different types of IR systems targeting to meet different users' information needs. For IR researchers who are interested in understanding the IR ranking models, what they want is the IR system where they can modify the ranking model of the system more easily and quickly, test it, iterate to next round. On the other hand, an IR system could also be mainly for end users. Users of the system do not necessarily know the details of how to design, how to implement the ranking algorithm of the system – they just need to know how to use the system to meet their information needs. One such example is the contextual suggestion system [59, 61, 62, 60, 65]. In our work we focus on two IR systems which are both related long-term information retrieval task. The first one is ranking model performance upper bound analysis using unified evaluation system which is mainly for

[1] http://www.internetlivestats.com/google-search-statistics/

IR researchers and the other is contextual suggestion which is mainly for end users. Specifically, we explorer the context and its impact for these two systems.

The word "*Context*" is originally defined as "the set of circumstances or facts that surround a particular event, situation, etc." In our work, context is extensively studied in order to reveal and quantify its impact for different research problems. We will show more details in the following.

1.1 Unified IR Evaluation System

The first problem we address is a long standing problem in evaluating the effectiveness of IR system[2]. For a typical IR evaluation system the ideal case is to have a unified testing environment which is responsible for everything related to the evaluation process except the ranking model part. That said, everything including pre-processing and indexing the documents, generating the ranking list, evaluating the results, the choice of evaluation metrics and interpretation of the performance, all should under the same setting if one's purpose is purely compare the effectiveness of different ranking models. For this problem the unified testing environment is regarded as the context of the evaluation process.

Apparently the context of this problem is the very basis of any kind of comparison between ranking models and thus should be carefully treated. Without this context researchers cannot make sound claim about their proposed models. Unfortunately, there is no such environment for the IR community. People continuously report different performances on the same baseline model [64] and this casts doubt on the real effectiveness of the proposed models.

In our work, two systems, namely VIRLab [16] and RISE [64] are proposed in order to offer a unified context to the IR community for standardization of comparing ranking models. The uniqueness and the advantage of these two systems is that

[2] There are several aspects in IR system can be evaluated. In our work, we focus on the evaluation of effectiveness of the system. Specifically only the effectiveness of the ranking model is investigated

they offer centralized and controlled IR evaluation systems which facilitate the fair comparison of retrieval models. The systems are the instantiation and expansion of Privacy Preserving Evaluation (PPE) [16] and Evaluation as a Service (EaaS) [48]. With the help of these systems (in our case we use RISE) we are able to conduct a comprehensive reproducibility study for information retrieval models. In particular, we implement and evaluate more than 20 basic retrieval functions over 16 standard TREC collections. Experimental results allow us to make a few interesting observations. We first compare the evaluation results with those reported in the original papers, and find that the performance differences between the reproduced results and the original ones are small for majority of the retrieval functions. Among all the implemented functions, only one of them consistently generates worse performance than the one reported in the original paper. Moreover, we report the retrieval performance of all the implemented retrieval functions over all the 16 TREC collections including recently released ClueWeb sets. To the best of our knowledge, this is the first time of reporting such a large scale comparison of IR retrieval models. Such a comparison can be used as the performance references of the selected models.

1.2 Boundary Theory of Bag-of-Terms Models

With the unified IR evaluation system like VIRLab and RISE at hand we are able to compare different ranking models. After an ever comprehensive comparison [64, 63] we find that the optimum performances of several TREC collections are quite similar for different models. For those models [49, 53, 68, 2, 17, 35, 21] all of them are based on bag-of-terms document representation assumption. That is, terms in the document are independent with each other and the occurrence (or absence) of one term does not affect the occurrence (or absence) of any other terms. For most of the models they mainly consist of basic signals (statistics) such as Term Frequency (TF), Inverted Document Frequency (IDF), Document Length Normalization (DLN) and other collection statistics [15]. We separate and organize these conditions and make them as the context of these ranking models since they are the theoretical foundation of these retrieval functions. With this context one interesting question here would be: it remains unclear whether we have reached the performance upper bound for such retrieval models. If so, what is the upper bound performance? If not, how can we do better?

To find the performance upper bound is quite challenging: although most of the IR ranking models deal with basic signals, how they combine the signals to compute the relevance scores are quite diverse due to different implementations of IR heuristics [15]. This kind of variants makes it difficult to generalize the analysis. Moreover, typically there are one or more free parameters in the ranking models which can be tuned via the training collections. These free parameters make the analysis more complicated. In our work, we simplify the problem and just focus on single-term queries and study how to estimate the performance bound for retrieval functions utilizing only basic ranking signals. With only one term in a query, many retrieval functions can be greatly simplified. For example, Okapi BM25 and Pivoted normalization functions have different implementations for the IDF part, but this part can be omitted in the functions for single-term queries because it would not affect the ranking of search results. All the simplified functions can then be generalized to a general function form for single-term queries. As a result, the problem of finding the upper bound of retrieval function utilizing basic ranking signals becomes that of finding the optimal performance of the generalized retrieval function. We propose to use cost/gain analysis to solve the problem [5, 6, 13]. As the estimated performance upper bound of simplified/generalized model is in general better than the existing ranking models, our finding provides the practical foundation of the potentially more effective ranking models for single term queries.

1.3 Contextual Suggestion

Another research endeavor of our work is to provide better IR system for end users. This kind of system is different from the previous one since end users do not necessarily know about the implementation or design details of the system – what they want is how to use the system to meet their information needs. Contextual suggestion is one of such example. The task of contextual suggestion is to recommend interesting venues to the users based on contextual information such as geographic location, temporal information and user's activity history. Traditionally context of this problem is often referred to the physical conditions of the users. In our work we argue that user's long-term, static preferences should also be included in the context as it is the key condition we have to rely on.

There are two necessary steps to tackle the contextual suggestion problem and both of them rely on the context we have defined: (1) identify a set of candidates that satisfy the contextual requirements, e.g., places of interest that are close to a target place; (2) rank the candidates with respect to the user interest. User profiling is the key component to effectively rank candidate places with respect to a user's information need and this is the reason we include the user preference history into the context.

In order to model use profile we first propose to leverage the category and description information about the places in user's activity history to construct user profiles [58]. The advantage of such approach is the ease of computation and the satisfactory results [10]. We further find that using category or description to build a user profile is not enough: category of places is too general to capture a user's underlying needs; while the text description of a place is too specific to be generalized to other places. In other studies [59, 61, 62, 60, 65] we leverage opinion, i.e. opinion ratings and the associated text reviews, to construct an opinionated user profile. By doing like this we aim to explain "why the user likes or dislikes the suggestion" instead of simply recording "what places the user liked or dislike" in the search history. The problem of this approach is that on-line opinions are notoriously skewed as only very small number of people post their opinions. To address this data sparsity challenge we propose to also include the opinions from similar users as the current user to construct the profile of current user. The assumption here is that users with similar ratings have the similar reasons of giving the rating. By modeling the candidate places in the similar fashion the similarity between user profile and candidates profile is used to rank the candidates. We tried different representations of the text reviews when modeling the profiles. We further apply Learning-to-Rank (LTR) method to the similarity scores for the ranking method. Experiment results on TREC collections and a self-crawled Yelp collection validate the effectiveness of our method.

1.4 Summary

Previous studies in IR rarely separated the context apart from other components of a specific research problem. However, we argue that the context of different IR systems or components should be carefully treated and extensively studied as the impact of the context is big enough to influence the results some IR studies, e.g. the ones aforementioned above.

As of future work, we would further explorer in two directions. The first one is to quantify the impact of the context of unified IR evaluation system. There is no previous work on how much difference does the usage of different retrieval tools bring. We hope to be the first to report on standardizing and quantifying the impact so that the IR community could be aware of such divergence and can better evaluate the contributions of using various tools. The second one is to provide more sound justification about the boundary theory of ranking model performance. Specifically, we want to extend the current analysis on the single term queries to multiple term queries. We also would like to try other method, e.g. explanatory analysis, to achieve the same goal.

The rest of the thesis is organized as follows. First, we discuss related work in Chapter 2. We describe our reproducibility study using VIRLab and RISE in Chapter 3. We explain how we use cost/gain analysis to find the performance upper bound for single term queries in Chapter 4. In Chapter 5, we investigate the problem of contextual suggestion and present our approaches. We then discuss future work in Chapter 6. Finally, we summarize the contributions of the thesis on Chapter 7.

Chapter 2

RELATED WORK

In this thesis, we investigate the impact of context in different IR systems. Specifically, two kinds of systems are mainly explored. The first one is the unified evaluation system where the context is the evaluation system itself on which the comparison of different ranking models relies. Based on the unified evaluation system we tried to provide analytical performance upper bound for bag-of-terms ranking models. The other system is the contextual suggestion. The key of solving this problem is to build the user profile based on the user preference history so we include user preference history as part of the context of this problem. Extensive work have been done on solving these research questions. We now survey the related work in the literature and discuss the differences with our proposed approaches in detail.

2.1 Unified IR Evaluation System

There have been significant efforts on developing various web services for IR evaluation. Lin et al. [32] proposed an open-source IR reproducibility challenge where they split the IR system into pieces of components such as two kinds of tokenization methods and four different IR toolkits. By easily configuring different combinations of these components, we can have a partially filled matrix indicating the performances of specific combinations of the components. Such transparent experiment set up makes it possible to have a better understanding about the impact of different components. Gollub et al. [18] described a reference implementation of their proposed IR evaluation web service which bears the important properties like web dissemination and peer-to-peer collaboration. Hanbury et al. [19] reviewed some of the existing automated

IR evaluation approaches and proposed a framework for web service based component-level IR system evaluation. Lagun and Agichtein proposed a web service, which enables large scale studies of remote users [28]. Their system focused on providing a platform that reproduces and extends the previous findings on how users interact with the search engine especially the search results.

Our developed *RISE* system is closely related to the ideas of Privacy Preserved Evaluation (PPE) [16] and Evaluation as a Service (EaaS) [48, 33, 34]. The system is designed as a web service to provide a unified interface for the users to evaluate their models/algorithms. This design enables the system to host the data collections instead of shipping the data collections to researchers, which can ensure the privacy of the collections. VIRLab [16] provides similar Web service for users to implement retrieval functions, but it is mainly designed to facilitate teaching IR models. Thus, it does not support as many collection statistics as those provided in the *RISE* system, and the users can not see the functions implemented by other users. The uniqueness of *RISE* system is that it is specifically designed to facilitate the implementation and evaluation of retrieval functions.

The SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR) [22] is one of the venues that encourage the study of reproducibility. Their reproducibility challenge invited developers of 7 open-source search engines to provide baselines for TREC GOV2 collection. Trotman et. al. [56] and Muhleisen el. al. [39] have also tried to reproduce retrieval results for IR models, but the number of retrieval functions and the number of collections used in these studies (1 function 1 collection for [56] and 9 functions 10 collections for [39]) are not as large as what we studied in this paper.

Compared with the previous studies, our work is different in the following two aspects. First, the *RISE* system is specifically designed for the reproducibility study of retrieval models. It hides details about collection processing and evaluation, and enables users to focus on only the implementation of retrieval models. Due to its flexibility, we are able to implement and compare a wide range of retrieval functions that were not implemented in any other open-source toolkits. Second, our reproducibility study includes more retrieval functions and more data collections. The ultimate goal of the *RISE* system is to provide a complete set of benchmark results of IR models.

2.2 Boundary Theory of Bag-of-Terms Models

Although there are lots of effective ranking models proposed by researchers, there are fewer studies dedicated to the theoretical analysis of their performances upper bound. One related domain is the constraint analysis [15] which proposes formal constraints that a reasonable ranking model should bear. Examples of the constraints including how should a ranking model incorporate TF, how to regulate the interaction of TF and DL, how to penalize long document in the collection, etc. The constraint analysis provides a general guide of how a reasonable ranking model should be designed. Our work further explores this direction by providing the practical performance upper bound as well as the optimal parameters which helps to fine tune the constraint theory.

Our estimation method is mostly inspired by the RankNet [6, 5] and the LambdaRank [5, 13] which are successful in the learning to rank domain. In their works they apply the pair-wise documents comparison for a specific query which is also adopted by our work. However, we did two different things in our work: (1) the aforementioned techniques apply neural network as the underlying model while we follow the rationale proposed by some classic ranking model, i.e. the ranking score should be positively correlated with TF and inversely correlated with DL, to find the local optimum of the generalized ranking models. (2) we aim to optimize MAP instead of NDCG and we proposed a simplified equation for calculating the difference of MAP if two documents are swapped in the ranking list which can make the analysis more efficiency. There is another work which indeed directly optimizes MAP called SVMMAP [67]. SVMMAP is actually another learning to ranking algorithm based on support vector machine. It performs optimization only on a working set of constraints which is extended with the most violated constraint at each optimization step. Taylor et al. [55] used the cost analysis to predicate a family of BM25 ranking models. They however did not apply the gain analysis which has shown to be superior in our experiments.

2.3 Contextual Suggestion

The problem of contextual suggestion was first introduced at TREC in 2012, and the track has been running in the past three years [10, 9]. Although the details of the track varied, the task remains the same. Given a user's preferences on a set of example suggestions and a context, track participants are expected to return a ranked list of new suggestions that are likely to satisfy both the user preferences (based on their preferences on the example suggestions) as well as the contexts such as geotemporal locations. Each example suggestion includes a title, description and an associated URL. For each user, we know their preferences on part or all of the example suggestions.

Most TREC participants retrieved candidate suggestions from various online services such as Google Place or Yelp based on the geographical context and then use some heuristics, e.g. nightclub will not be shown if the temporal context is in the morning, to filter out the suggestions that do not match the temporal contexts [10, 9]. After that, the task is to retrieve useful suggestions based on user preferences. Most participants formulated the task as a content-based recommendation problem [23, 66, 58, 47, 50, 25, 61, 57, 30, 31, 38]. A common strategy adopted by top-ranked participants of TREC is to estimate a user profile based on the example suggestions and then rank candidate suggestions based on their similarities to the user profile. The basic assumption is that a user would prefer suggestions that are similar to those example suggestions liked by the user. Various types of information about the suggestions have been used to estimate user profiles which include the description of the places [58, 23, 25], the categories of the places [31, 30, 38, 66, 27], and the web sites of the places [58, 23, 25].

Specifically, many studies used terms from the description of the places or the web pages of the example suggestions to construct user profiles, and then various similarity measures are used to rank the candidates [58, 23, 25]. A few studies also explored the use of category information for user profiling and candidate ranking. For example, Li and Alonso [31] utilized the accumulative category scores to model both user and candidate profiles, and then use the full range cosine similarity between the two profiles for candidate ranking. Li et al. [30] leveraged how likely each popular category is liked/disliked by users to construct user profiles, and the candidate ranking is to favor suggestions from a user's favorite categories. McCreadie et al. [38] proposed to rank the candidates by comparing two trees of finer-grained categories between user profile and candidate profile using a tree-matching technique. Diversification is then applied so that the categories of top ranked candidates are normalized. Yates et al. [66] proposed to recommend the candidates which are proportional to the number of example suggestions in each category. Koolen et al. [27] applied a similar method with a major modification of retrieving the category information from Wikitravel[1].

However, none of other groups has tried to leverage the reviews about these places to estimate the user profile as what we propose in this paper. As we mentioned earlier, using either descriptions or categories can not precisely capture what a user likes or dislikes. However, online reviews offer rich information about user opinions and should be leveraged in user profiling. To the best of our knowledge, we are the first ones who incorporate opinions as user profiles in pursuing better solution for contextual suggestion.

2.3.1 Recommendation Systems

The problem of contextual suggestion is also similar to collaborative filtering [54]. Collaborative filtering assumes that similar users would share similar ratings, and focuses on predicting the user rating based on such an assumption. It often requires a large number of past user preferences to be more accurate and sometimes it may suffer from data sparsity problem which is known as the cold start problem [52]. In order to solve the data sparsity problem, reviews were incorporated to improve the performance. Hariri et al. [20] inferred the context or the intent of the trip by analyzing reviews.

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

In particular, they used latent Dirichlet Allocation to identify the topics from the reviews, and the final ranking scores are generated based on both the context scores as well as the scores generated by traditional collaborative filtering methods. Jakob et al. [24] proposed to cluster the features and then apply natural language processing techniques to identify the polarity of the opinions. A few studies also focused on leveraging Location Based Social Network to solve the data sparsity problem. Noulas et al. [40] applied random walk based on latent space models and computed a variety of similarity criteria with venue's visit frequencies on the location based social newtowkr. Bao et al [4] proposed to first constructing a weighted category hierarchy and then identify local experts for each category. The local experts are then matched to a given user and the score of the candidate is inferred based on the opinions of the local experts.

Our work is also related to other studies that utilized reviews to improve the performance of recommendation systems [44, 51, 46, 20, 29]. Raghavan et al. [46] proposed to use the helpfulness, features from the text reviews and the meta-data (average rating, average length of text reviews and etc.) of the opinions to train a regression model in order to generate a quality score for each opinion. The quality score is then incorporated into the probabilistic matrix factorization as an inverse factor which affects the variance of the prediction from the mean of the factor model. Levi et al. [29] extended this study and analyzed the review texts to get the intent, features and the ratings for each feature. Qumsiyeh and Ng [44] explored the aspects in the reviews and computed the probability of each genres (categories) in each rating level. Their work is limited to the applications in multimedia domains, and the genres of each type of media is pre-defined.

Our work is different from these previous studies in the following aspects. First, our focus is to directly use reviews to model user profile while previous studies mainly used reviews to predict the rating quality or the user intent. Second, existing studies on collaborative filtering were often evaluated on only specific applications, e.g., movies, hotels, and it is unclear how those methods could be generalized to other domains. In contrast, our proposed method is not limited to any specific domains and can be applied to a more general problem set up.

2.3.2 Text Summarization

The summary generation of our work is related to automatic text summarization. Automatic text summarization has been well studied for traditional documents such as scientific documents and news articles [45]. In particular, previous work has studied various problems in this area including extractive summarization, abstractive summarization, single-document summarization and multiple-document summarization [8]. More recently, there have been effort on opinion mining and summarization [43, 41, 42, 37, 36, 26, 12, 3, 7, 11, 14]. Most of them involve in the finer partition of the reviews and polarity judging of each partition. Common strategies include part-of-speech analysis, negation identification and etc. Unlike the previous effort, we focus on generating a *personalized* summary for a suggestion. Since the information about the suggestion is scattered in many places, including description, web sites and reviews, the summarization needs to synthesize the information from these heterogeneous information sources. Instead of extracting the information from a single source, we try to leverage one information source to guide the extractive summarization process in other sources and then assemble all the extracted summaries together into a *structural* way. Another main difference of our work from previous studies is to utilize the user profile to generate personalized summaries.

Chapter 3

REPRODUCIBILITY STUDY USING UNIFIED IR EVALUATION SYSTEM

Chapter 4

BOUNDARY THEORY OF IR RANKING MODELS

Chapter 5

CONTEXTUAL SUGGESTION

The increasing use of mobile devices enables an information retrieval (IR) system to capitalize on various types of contexts (e.g., temporal and geographical information) about its users. Combined with the user preference history recorded in the system, a better understanding of users' information need can be achieved and it thus leads to improved user satisfaction. More importantly, such a system could *proactively* recommend suggestions based on the contexts.

User profiling is essential in contextual suggestion. However, given most users' observed behaviors are sparse and their preferences are latent in an IR system, constructing accurate user profiles is generally difficult. In our work, we focus on location-based contextual suggestion and propose two approaches to construct user profiles.

The first approach uses the categories and/or descriptions from users' activities history to build user profile. The rationale here is that users are at a better chance to favor the places that are similar to what she liked before in terms of the category/description of the places. In reality, one user would typically have several positively rated and also several negatively rated suggestions in the past. We compute the similarity of category/description between each candidate suggestion and all places in the user's activity history and combine the averages of both positive and negative scores. Experiment results show pretty decent performance of using this method.

The second approach leverages the users' opinions to form the user profiles. By assuming users would like or dislike a place with similar reasons, we construct the opinion-based user profile in a collaborative way: opinions from the other users are leveraged to estimate a profile for the target user. Candidate suggestions are represented in the same fashion and ranked based on their similarities with respect to the user profiles. Moreover, we also develop a novel summary generation method that utilizes the opinion-based user profiles to generate personalized and high-quality summaries for the suggestions. Experiments conducted over three standard TREC Contextual suggestion collections and a Yelp data set show the advantage of this approach and the system developed based on the proposed methods have been ranked as top 1 in both TREC 2013 and 2014 Contextual Suggestion tracks.

5.1 Introduction

The increasing availability of internet access on mobile devices, such as smart phones and tablets, has made mobile search a new focus of information retrieval (IR) research community. The contextual information such as geographical and temporal information that is available in mobile search environment provides unique opportunities for IR systems to better understand its users. Moreover, a user's preference history collected in a mobile search system can be incorporated with such contextual information to better understand the user's informational need. Ideally, a mobile search system should thus *proactively* generate suggestions for various user information needs. For example, it would be useful to automatically send recommendations about the Beatles museum to a music fan who travels to Liverpool. In addition to returning a list of suggestions to the user, it would also be useful to provide a short yet *informative summary* for each suggestion so that the user can easily decide whether the recommended suggestion is interesting before accepting it. This problem is referred to as *contextual suggestion*, and has been identified as one of the IR challenges (i.e,. "finding what you need with zero query terms") in the SWIRL 2012 workshop [1].

5.2 Problem Formulation

The problem of contextual suggestion can be formalized as follows. Given a user's contexts (e.g., location and time) and the her/his preferences on a few example suggestions, the goal is to retrieve candidate suggestions that can satisfy the user's information need based on both the context and preferences. For each returned candidate suggestion, a short description may also be returned so that the user could decide whether the suggestion is interesting without going to its website. For example, assume that a user liked "Magic Kingdom Park" and "Animal Kingdom", but disliked "Kennedy Space Center". If the user is visiting Philadelphia on a Saturday, the system is expected to return a list of suggestions such as "Sesame Palace" together with a short summary of each suggestion, e.g., "Sesame Place is a theme park in Langhorne, Pennsylvania based on the Sesame Street television program. It includes a variety of rides, shows, and water attractions suited to very young children."

Since our paper focuses on user modeling, we assume that we have filtered out the suggestions that do not meet the context requirement and the remaining suggestions only need to be ranked based on the relevance to user preferences. Note that the filtering process based on contexts can be achieved by simply removing the suggestions that do not satisfy the contextual requirements, such as the ones that are either too far away from the current location or those that are currently closed.

The remaining problem is essentially a ranking problem, where candidate suggestions need to be ranked based on how relevant the suggestions are with respect to a user's interest. Formally, let U denote a user and CS denote a candidate suggestion, we need to estimate $S(U, CS)$, i.e., the relevance score between the user and the suggestion.

It is clear that the estimation of the relevance score is related to how to represent U and CS based on the available information. Let us first look at what kind of information we can gather for U and CS. For each user U, we know the user's preferences (i.e., ratings) for a list of example suggestions. We denote an example suggestion ES and its rating given by user U as $R(U, ES)$. For a suggestion (either CS or ES), we assume that the following information about the suggestion is available: the text description such as title and category and online opinions about this suggestion. Note all the information can be collected from online location services such as Yelp and Tripadvisor.

5.3 Method with Category and Description

5.3.1 Gathering Candidate Suggestions

To collect candidate suggestions, we crawl the information from multiple online sources such as Yelp and Foursquqre based on the geographical information from the 50 contexts. The collected candidate suggestions have different fields such as categories, web sites, business hours, etc. Moreover, we also crawl the web site for each candidate suggestion.

5.3.2 Ranking based on User Profiles

We now describe how to rank candidate suggestions based on user profiles. The profile of each user consists of the user's preferences for 49 example locations. The locations that a user likes are referred to as "positive examples", and those disliked by the user are referred to as "negative examples". Intuitively, the relevance score of a candidate suggestion should be higher when it is similar to positive examples while different from the negative examples.

Formally, we denote U as a user and C as a candidate suggestion. Moreover, let $P(U)$ denote positive examples, i.e., a set of places that the user likes, and $N(U)$ denote negative examples, i.e., a set of places that the user dislikes. The relevance score of C with respect to U can then be computed as follows:

$$\begin{aligned} S(U,C) &= \varphi \times S_P(P(U),C) + (1-\varphi) \times S_N(N(U),C) && (5.1)\\ &= \varphi \times \frac{\sum_{p \in P(U)} SIM(p,C)}{|P(U)|} + (1-\varphi) \times \frac{\sum_{p \in N(U)} SIM(n,C)}{|N(U)|}, && (5.2) \end{aligned}$$

where $\varphi \in [0,1]$ and it regularize the weights between the positive and negative examples. When $\varphi = 1$, the highly ranked suggestions would be those similar to the locations that the user likes. When $\varphi = 0$, the highly ranked suggestions would be those different from the locations that the user dislikes. $S_P(P(U),C)$ measures the similarity between the positive user profile and the candidate suggestion, and we assume that it can be computed by averaging the similarity scores between each positive example and

Table 5.1: **Examples of Categories in Example Suggestions**

NAME	**Yelp Categories**	**FourSquare Categories**

HoSu Bistro	SushiRestaurant→Restaurants;
	KoreanRestaurant→Restaurants;
	JapaneseRestaurant→Restaurants
		SushiRestaurant→Food

The Rex	JazzBlues→MusicVenues→Arts	JazzClub → MusicVenue → ArtsEntertainment
St. Lawrence Market	Grocery→Shopping; FarmersMarket→Shopping	FarmersMarket → FoodDrinkShop → ShopService
...	...	...

the candidate suggestion. $|P(U)|$ corresponds to the number of positive examples in the user profile.

Thus, it is clear that the problem of computing the relevance score of a candidate suggestion with respect a user can be boiled down to the problem of computing the relevance score between a candidate suggestion and a place mentioned in the user profile, i.e., $SIM(e, C)$, where e is an example from the user profile. We explore the following two types of information to compute $SIM(e, C)$: (1) the category of a place; and (2) the description of a place.

5.3.2.1 Category-based similarity:

Category is a very important factor that may greatly impact user preferences. The categories of the crawled suggestions are often hierarchical. Here is an example category, i.e., *[History Museum→ Museum→Arts].* The categories becomes more general from the left to the right. In this example, *Arts* is the most general category while *History Museum* is the most specific category. Note that we represent the hierarchical categories as a set of categories in this paper.

We can compute $SIM(e, C)$ based on the category similarities between e and C as follows:

$$SIM_{\mathcal{C}}(e, C) = \frac{\sum_{c_i \in \mathcal{C}(e)} \sum_{c_j \in \mathcal{C}(C)} \frac{|Intersection(c_i, c_j)|}{max(|c_i|, |c_j|)}}{|\mathcal{C}(e)| \times |\mathcal{C}(C)|}, \tag{5.3}$$

where $\mathcal{C}(e)$ denotes the set of categories of location e and $|Intersection(c_i, c_j)|$ is the number of common categories between c_i and c_j. Recall that we crawled the candidate suggestions from two online sources. Table 5.1 shows example categories from both sources. Thus, we combine the similarity scores computed based on the categories from them as follows:

$$SIM_{\mathcal{C}'}(e, C) = \phi \times SIM_C^{Yelp}(e, C) + (1 - \phi) \times SIM_C^{FourSquare}(e, C). \tag{5.4}$$

In our experiments, we set ϕ as 0.5 which means the importance of category score of Yelp and FourSquare are the same.

5.3.2.2 Description-based similarity:

In example suggestions, each suggestion has its unique description which typically is at a short length. We want to learn how the descriptions can affect people's decision on different places. By comparing the descriptions of training suggestions with textual web sites of testing suggestions we may find some interesting connections. We use textual web sites of testing suggestions because we believe that textual web sites are more reliable than descriptions especially when we rank candidate suggestions. The similarity used function is the F2EXP ranking function [17]. Thus, we compute the similarity scores as follows: $SIM_{\mathcal{D}}(e, C) = F2EXP(DES(e), DES(C))$, where $DES(e)$ is a description of the example place e.

5.3.3 Ranking the candidates based on the contexts

There are two types of context: geographical and temporal information. All of the candidate suggestions crawled in the first step are related to the geographical requirement because they are retrieved based on it. To make sure the suggestions satisfying the temporal requirement, we collected the business hours from Yelp, and then assign the business hours for each cateogry if the majority suggestions in that category follow the same business hour. Candidate suggestions that do not meet the temporal requirement are then removed from the final ranking list.

Chapter 6

FUTURE WORK

BIBLIOGRAPHY

[1] James Allan, Bruce Croft, Alistair Moffat, and Mark Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. *SIGIR Forum*, 46(1):2–32, 2012.

[2] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010.

[4] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 199–208, New York, NY, USA, 2012. ACM.

[5] Christopher J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, June 2010.

[6] C.J.C. Burges, R. Ragno, and Q.V. Le. Learning to rank with non-smooth cost functions. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, January 2007.

[7] Hsinchun Chen and David Zimbra. Ai and opinion mining. *IEEE Intelligent Systems*, 25(3):74–80, May 2010.

[8] Dipanjan Das and Andre F. T. Martins. A survey on automatic text summarization. 2007.

[9] Adriel Dean-Hall, Charles Clarke, Jaap Kamps, Paul Thomas, Nicole Simone, and Ellen Voorhees. Overview of the trec 2013 contextual suggestion track. In *Proceedings of TREC'13*, 2013.

[10] Adriel Dean-Hall, Charles Clarke, Jaap Kamps, Paul Thomas, and Ellen Voorhees. Overview of the trec 2012 contextual suggestion track. In *Proceedings of TREC'12*, 2012.

[11] Lipika Dey and SK Mirajul Haque. Opinion mining from noisy text data. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):205–226, 2009.

[12] Xiaowen Ding and Bing Liu. The utility of linguistic rules in opinion mining. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812. ACM, 2007.

[13] Pinar Donmez, Krysta M. Svore, and Christoper J.C. Burges. On the local optimality of lambdarank. In *SIGIR*. Association for Computing Machinery, Inc., July 2009.

[14] Andrea Esuli. Automatic generation of lexical resources for opinion mining: models, algorithms and applications. *SIGIR Forum*, 42(2):105–106, 2008.

[15] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 49–56, New York, NY, USA, 2004. ACM.

[16] Hui Fang, Hao Wu, Peilin Yang, and ChengXiang Zhai. Virlab: A web-based virtual lab for learning and studying information retrieval models. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 1249–1250, New York, NY, USA, 2014. ACM.

[17] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM.

[18] Tim Gollub, Benno Stein, and Steven Burrows. Ousting ivory tower research: Towards a web framework for providing experiments as a service. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1125–1126, New York, NY, USA, 2012. ACM.

[19] Allan Hanbury and Henning Müller. Automated component-level evaluation: Present and future. In *Proceedings of the 2010 International Conference on Multilingual and Multimodal Information Access Evaluation: Cross-language Evaluation Forum*, CLEF'10, pages 124–135, Berlin, Heidelberg, 2010. Springer-Verlag.

[20] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng. Context-aware recommendation based on review mining. In *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, 2011.

[21] Ben He and Iadh Ounis. A study of the dirichlet priors for term frequency normalisation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 465–471, New York, NY, USA, 2005. ACM.

[22] Frank Hopfgartner, Allan Hanbury, Henning Müller, Noriko Kando, Simon Mercer, Jayashree Kalpathy-Cramer, Martin Potthast, Tim Gollub, Anastasia Krithara, Jimmy Lin, Krisztian Balog, and Ivan Eggel. Report on the evaluation-as-a-service (eaas) expert workshop. *SIGIR Forum*, 49(1):57–65, June 2015.

[23] Gilles Hubert and Guillaume Cabanac. Irit at trec 2012 contextual suggestion track. In *Proceedings of TREC'12*, 2012.

[24] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, pages 57–64, New York, NY, USA, 2009. ACM.

[25] Ming Jiang and Daqing He. Pitt at trec 2013 contextual suggestion track. In *Proceedings of TREC'13*, 2013.

[26] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[27] Marijn Koolen, Hugo Huurdeman, and Jaap Kamps. University of amsterdam at the trec 2013 contextual suggestion track: Learning user preferences from wikitravel categories. In *Proceedings of TREC'13*, 2013.

[28] Dmitry Lagun and Eugene Agichtein. Viewser: Enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 365–374, New York, NY, USA, 2011. ACM.

[29] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *Proceedings of the RecSys'12*, 2012.

[30] Hanchen Li, Zhen Yang, Yingxu Lai, Lijuan Duan, and Kefeng Fan. Bjut at trec 2014 contextual suggestion track: Hybrid recommendation based on open-web information. In *Proceedings of TREC'14*, 2014.

[31] Hua Li and Rafael Alonso. User modeling for contextual suggestion. In *Proceedings of TREC'14*, 2014.

[32] Jimmy Lin, Matt Crane, Andrew Trotman, Jamie Callan, Ishan Chattopadhyaya, John Foley, Grant Ingersoll, Craig MacDonald, and Sebastiano Vigna. Toward reproducible baselines: The open-source ir reproducibility challenge. In Nicola Ferro, Fabio Crestani, Marie-Francine Moens, Josiane Mothe, Fabrizio Silvestri, Giorgio Maria Di Nunzio, Claudia Hauff, and Gianmaria Silvello, editors, *ECIR*, volume 9626 of *Lecture Notes in Computer Science*, pages 408–420. Springer, 2016.

[33] Jimmy Lin and Miles Efron. Evaluation as a service for information retrieval. *SIGIR Forum*, 47(2):8–14, January 2013.

[34] Jimmy Lin and Miles Efron. Infrastructure support for evaluation as a service. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 79–82, New York, NY, USA, 2014. ACM.

[35] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 7–16, New York, NY, USA, 2011. ACM.

[36] Inderjeet Mani. *Automatic summarization*, volume 3. 2001.

[37] Inderjeet Mani and Mark T Maybury. *Advances in automatic text summarization.* 1999.

[38] Richard McCreadie, Romain Deveaud, M-Dyaa Albakour, Stuart Mackie, Nut Limsopatham, Craig Macdonald, Iadh Ounis, Thibaut Thonet, and Bekir Taner. University of glasgow at trec 2014: Experiments with terrier in contextual suggestion, temporal summarisation and web tracks. In *Proceedings of TREC'14*, 2014.

[39] Hannes Mühleisen, Thaer Samar, Jimmy Lin, and Arjen de Vries. Old dogs are great at new tricks: Column stores for ir prototyping. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 863–866, New York, NY, USA, 2014. ACM.

[40] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*, SOCIALCOM-PASSAT '12, pages 144–153, Washington, DC, USA, 2012. IEEE Computer Society.

[41] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010.

[42] Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004.

[43] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[44] Rani Qumsiyeh and Yiu-Kai Ng. Predicting the ratings of multimedia items for making personalized recommendations. In *Proceedings of SIGIR'12*, 2012.

[45] D. Radev, H. Jing, and M. Budzikowska. Introduction to the special issue on summarization. *Computational Linguistics*, 28(4):399.

[46] Sindhu Raghavan, Suriya Gunasekar, and Joydeep Ghosh. Review quality aware collaborative filtering. In *Proceedings of RecSys'12*, 2012.

[47] Ashwani Rao and Ben Carterette. Udel at trec 2012. In *Proceedings of TREC'12*, 2012.

[48] Jinfeng Rao, Jimmy Lin, and Miles Efron. Reproducible experiments on lexical and temporal feedback for tweet search. In Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr, editors, *Advances in Information Retrieval*, volume 9022 of *Lecture Notes in Computer Science*, pages 755–767. Springer International Publishing, 2015.

[49] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1996.

[50] Dwaipayan Roy, Ayan Bandyopadhyay, and Mandar Mitra. A simple context dependent suggestion system. In *Proceedings of TREC'13*, 2013.

[51] Jose San Pedro, Tom Yeh, and Nuria Oliver. Leveraging user comments for aesthetic aware image search reranking. In *Proceedings of WWW'12*, 2012.

[52] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[53] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

[54] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, January 2009.

[55] Michael Taylor, Hugo Zaragoza, Nick Craswell, Stephen Robertson, and Chris Burges. Optimisation methods for ranking functions with multiple parameters. CIKM '06, pages 585–593, New York, NY, USA, 2006. ACM.

[56] Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, Melbourne, VIC, Australia, 2014. ACM.

[57] Di Xu and Jamie Callan. Modelling psychological needs for user-dependent contextual suggestion. In *Proceedings of TREC'14*, 2014.

[58] Peilin Yang and Hui Fang. An exploration of ranking-based strategy for contextual suggestion. In *Proceedings of TREC'12*, 2012.

[59] Peilin Yang and Hui Fang. An opinion-aware approach to contextual suggestion. In *Proceedings of TREC'13*, 2013.

[60] Peilin Yang and Hui Fang. Opinion-based user profile modeling for contextual suggestions. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, pages 18:80–18:83, New York, NY, USA, 2013. ACM.

[61] Peilin Yang and Hui Fang. Exploration of opinion-aware approach to contextual suggestion. In *Proceedings of TREC'14*, 2014.

[62] Peilin Yang and Hui Fang. Combining opinion profile modeling with complex contextfiltering for contextual suggestion. In *Proceedings of TREC'15*, 2015.

[63] Peilin Yang and Hui Fang. Estimating retrieval performance bound for single term queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 237–240, New York, NY, USA, 2016. ACM.

[64] Peilin Yang and Hui Fang. A reproducibility study of information retrieval models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 77–86, New York, NY, USA, 2016. ACM.

[65] Peilin Yang, Hongning Wang, Hui Fang, and Deng Cai. Opinions matter: a general approach to user profile modeling for contextual suggestion. *Information Retrieval Journal*, 18(6):586–610, 2015.

[66] Andrew Yates, Dave DeBoer, Hui Yang, Nazli Goharian, Steve Kunath, and Ophir Frieder. (not too) personalized learning to rank for contextual suggestion. In *Proceedings of TREC'12*, 2012.

[67] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. SIGIR '07, pages 271–278, New York, NY, USA, 2007. ACM.

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