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 5

2 RELATED WORK **7**

3 REPRODUCIBILITY STUDY USING UNIFIED IR EVALUATION SYSTEM **8**

4 BOUNDARY THEORY OF IR RANKING MODELS **9**

5 CONTEXTUAL SUGGESTION **10**

5.1 Introduction 11
5.2 Problem Formulation 11
5.3 Method with Category and Description 13

5.3.1 Gathering Candidate Suggestions 13
5.3.2 Ranking based on User Profiles 13

5.3.2.1 Category-based similarity: 14
5.3.2.2 Description-based similarity: 15

5.3.3 Ranking the candidates based on the contexts 15

6 FUTURE WORK **17**

BIBLIOGRAPHY **18**

LIST OF TABLES

5.1 **Examples of Categories in Example Suggestions** 14

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. For IR researchers they want an IR system which is more fundamental, transparent and touchable. Such system could offer the researchers the ability to easily modify, test and iterate their hypothesis. 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.

In order to make better 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.

Previously, the typical work-flow of proposing a new IR ranking model is to implement the algorithm using the most convenient tool (e.g. Indri) of the researcher's choice. The ranking model is then evaluated against standard data collections and the performances are compared with commonly used baselines (e.g. BM25 [18]) and the advantage of the proposed model is presented. The problem of such methodology is that different tools and experimental settings may result in different results for the same model [27] and this is a long standing problem of IR community for over 20 years [3]. The fact casts doubts on the real effectiveness of the proposed ranking models and 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 easy-to-use IR evaluation system where user of the system can implement her/his our model without considering too much about the underlying framework. A web based Reproducible IR Evaluation System (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 experiment based on RISE serves as the the performance reference of most widely used IR models.

Another example is the boundary theory of ranking model performance. The context of this line of research is the bag-of-terms representation of the document assumption and the widely used statistics, e.g. Term Frequency (TF) and Inverted Document Frequency (IDF) that the ranking model consists of. We first compare the optimal performances of state-of-the-art ranking models and find the optimums are similar for those models even the underlying theories are different. To dig it deeper we use the cost/gain analysis which is commonly used in learning-to-rank (LTR) regime to find the optimum of single term queries for a family of ranking models that share similar strategy of combining key signals. 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.

The third domain is to investigate the usage of opinion to contextual suggestion. It is commonly agreed that how to model user profile is the key to the solution. We first model the user profile using the venue's category and description of user activity history. We further improve the method by leveraging the opinions from user's activity history to model the user profile. Such methodology naturally utilizes the rich text associated with the users which naturally makes the problem as a retrieval problem. 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 reveals the advantage of our method.

For the future work, there are mainly two directions we would like to explore

more deeper. 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, Microsoft Bing 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. The primary goal of IR research is to improve the effectiveness or the efficiency or both of the textual retrieval system. There are many related works dedicated to this line of research already [18, 19, 29, 11, 21, 14].

The word "*Context*" is originally defined as "the set of circumstances or facts that surround a particular event, situation, etc." Context always play important role in IR research. It is even the fundamental basis of some very important IR research efforts. The context in IR could be the experiment settings. It can also be the assumptions on which the ranking model relies. In our work, we pick three domains to show the existence and the impact of the context.

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

1.1 Unified IR Evaluation System

The first domain is related to the evaluation of IR system. There are many aspects that an IR system can be evaluated. In our work we focus on evaluating the effectiveness of the system. Many different techniques can be applied to address the effectiveness of the system. For example, Natural Language Processing (NLP) techniques [20, 16]. Topic Modeling [4, 13]. But most of the previous works target on the simple yet effective ranking models which usually applied to the document index. 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, ranking the documents, evaluating the results, the choice of evaluation metrics and interpretation of the performance should under the same setting if one's purpose is purely compare the effectiveness of different ranking models. Here the unified testing environment can be regarded as the context of the evaluation process.

The context here is the basis of any kind of comparison between ranking models. Without the unified testing environment people 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 [27] and this casts doubt on the real effectiveness of the proposed models.

In our work, two systems, namely VIRLab [10] and RISE [27] are proposed to specifically address the problem. The uniqueness and the advantage of these two system is that 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)[10] and Evaluation as a Service (EaaS)[17]. With the help of these systems (especially 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

Classic IR ranking models [18, 19, 29, 2, 11, 15, 12] are mainly based on bag-of-terms document representation assumption and they mainly consist of basic signals (statistics) such as Term Frequency (TF), Inverted Document Frequency (IDF), Document Length Normalization (DLN) and other collection statistics [9]. For this domain we can view the bag-of-terms assumption and the commonly used statistics as the context of the ranking models since they are the theoretical foundation of these retrieval functions. Under this context our question is whether we have reached the performance upper bound for retrieval functions using only basic ranking signals. 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 [9]. 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, 8]. 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

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. Context again, as the name of the problem shows, highlights this direction of research effort.

There are two necessary steps to tackle the contextual suggestion problem: (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. In our work we focus on the second problem assuming the first requirement has already been fulfilled. User profiling is the key component to effectively rank candidate places with respect to a user's information need and the question for us is how to effectively model the user profile? We first propose to leverage the category and description information about the places in user's activity history to construct user profiles [22]. The advantage of such approach is the ease of computation and the satisfactory results [7]. 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 [23, 25, 26, 24, 28] 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 thesuggestion" 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 question the foundation of 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

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:

$$S(U,C) = \varphi \times S_P(P(U),C) + (1-\varphi) \times S_N(N(U),C) \quad (5.1)$$

$$= \varphi \times \frac{\sum_{p \in P(U)} SIM(p,C)}{|P(U)|} + (1-\varphi) \times \frac{\sum_{p \in N(U)} SIM(n,C)}{|N(U)|}, \quad (5.2)$$

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 [11]. 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] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 601–610, New York, NY, USA, 2009. ACM.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[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] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[14] Xitong Liu and Hui Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[15] 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.

[16] Rada F. Mihalcea and Dragomir R. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[17] 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.

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

[19] 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.

[20] Ellen M. Voorhees. Natural language processing and information retrieval. In *Information Extraction: Towards Scalable, Adaptable Systems*, pages 32–48, London, UK, UK, 1999. Springer-Verlag.

[21] Hao Wu and Hui Fang. An incremental approach to efficient pseudo-relevance feedback. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 553–562, New York, NY, USA, 2013. ACM.

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

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

[24] 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.

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

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

[27] 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.

[28] 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.

[29] 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.