TRINITY COLLEGE DUBLIN

Revisiting Contextual Recommendation from an Information Retrieval Standpoint

Author:
Anirban CHAKRABORTY

Supervisor(s):
The late Prof. Séamus LAWLESS
Prof. Owen CONLAN

A thesis submitted in fulfilment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in

School of Computer Science and Statistics
Trinity College Dublin

MARCH, 2021

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signature ____________________ Date __________

Anirban Chakraborty

To my parents, and the loving memory of Shay.

Acknowledgements

First and foremost I would like to extend my most profound and sincere gratitude towards my late supervisor, Prof. Séamus (Shay) Lawless for introducing me to this research topic and providing his valuable guidance and unfailing encouragement. Prof. Lawless gave me an opportunity to pursue and explore an ambition I have held since my childhood. He was so supportive, a wonderful educator, a good friend, and an exemplary human being. I would like to dedicate this thesis to the loving memory of Shay, who suffered a tragic death after fulfilling his dream of scaling Mt. Everest in May 2019.

I also wish to thank Prof. Owen Conlan, who kindly stepped in to guiding me through the final stages of my PhD after Shay unexpectedly passed away.

I am deeply indebted to Dr. Debasis Ganguly for helping me to understand Information Retrieval (IR) research in general. Much of the work in this thesis would not have materialized without the support from Debasis, who always provided insightful ideas, constant support and inspiration throughout the course of this thesis. I enjoyed all the discussions with Debasis which were so fruitful that they eventually led to a number of publications. I am also thankful to both Debasis and my friend Dr. Dwaipayan Roy for helping me to understand relevance model, and helping me with Lucene.

Thanks to Dr. Annalina Caputo who was a postdoctoral researcher during the early years of my PhD under the supervision of Shay, and Dr. Mostafa Bayomi for believing in my work and offering me support, and guidance when I needed it. Thanks to Prof. Gareth J. F. Jones, who provided insightful suggestions on my PhD confirmation report, much of which is incorporated into this thesis. I would like to express a big thanks to my lab members (past and present) and friends at Trinity College Dublin and/or ADAPT centre, just to name a few - Dr. Gary Munnelly, Dr. Brendan Spillane, Dr. Yu Xu, Esraa Ali, Hao Wu, Procheta Sen, Dr. Harshvardhan J. Pandit, Dr. Nicole Basaraba, Dr. Lucy McKenna, Ramisa Hamed, Kieran Fraser, Dr. Aimee Borda, Dr. Aonghus McGovern, Sinéad Impey among many others for their support, friendship, and interest in my work and for some of the oddest conversations during the tea breaks. Thanks to Robbie, Kristina, Monica, and Jenny for miscellaneous administrative support.

I remember the old days in the IR lab at Indian Statistical Institute, Kolkata where I got to know about IR for the first time in my life. After finishing my master's thesis under the supervision of Prof. Swapan Kumar Parui, I started working on a couple of interesting IR tasks in the same lab. Thanks to Prof.

Parui for his sharp insight and guidance. I also got immense inspiration, and support from Dr. Mandar Mitra, Dr. Prasenjit Majumder, Dr. Jiaul Hoque Paik, and Dr. Kripabandhu Ghosh. In fact, the working partnership with Kripa was so strong that we had a number of publications even before I started my PhD. Also thanks to Dipasree, Ayan, Abhisek, Anabik, and Chandan for their encouragement. Thanks to Prof. Utpal Roy who introduced me to Prof. Parui, and it all started.

I am privileged to be in touch with many IR doyens even before I moved to Ireland, thanks to Forum for Information Retrieval Evaluation (FIRE). I had the opportunity to discuss some ideas with Prof. Jaap Kamps, Prof. Doug Oard, and Prof. Soumen Chakrabarti when I met them in SIGIR 2018.

I will be eternally grateful to my parents Nilratan Chakraborty, and Dipali Chakraborty for their unconditional love and care in every aspect of my life. I would not have come this far without their support. A special thanks to my friend Shainy Ojha. Despite the pressure of her own PhD, Shainy was so supportive in my struggles to help me survive this journey. I am grateful for her presence when it mattered the most.

I am thankful to ADAPT Centre for Digital Content Technology, which is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund, to support this research.

After Shay's untimely passing, I have had a rocky patch of time over 2019-20, which worsened with the Covid-19 pandemic. I am grateful to everyone who was there to have my back, in this unprecedented situation.

Thank you all.

Abstract

The challenge of providing personalized and contextually appropriate recommendations to a user is faced in a range of use-cases, e.g. recommendations for movies, places to visit, articles to read etc. In this thesis, we focus on one such application, namely that of suggesting 'points of interest' (POIs) to a user given her current context(s), by leveraging relevant information from her past preferences. An automated contextual recommendation algorithm is likely to work well if it can extract information from the preference history of a user (*exploitation*) and effectively combine it with information from the user's current context (*exploration*) to predict a POI's 'appropriateness' in the current context. To balance this trade-off between *exploitation* and *exploration*, we propose a generic unsupervised framework involving a factored relevance model (FRLM), comprising two distinct components, one corresponding to the historical information from past contexts, and the other pertaining to the information from the current context.

We further generalize the proposed model FRLM by incorporating the semantic relationships between terms in POI descriptors with the help of kernel density estimation (KDE) on embedded word vectors. Additionally, we show that trip-qualifiers, such as *trip-type* (e.g. vacation, work etc.) and *accompanied-by* (e.g. solo, friends, family etc.) are potentially useful sources of information that could be used to improve the effectiveness of POI recommendation in a current context (with a given set of these constraints). Using such information is not straight forward since users' text reviews of POIs visited in the past typically do not explicitly contain such annotations (e.g. a positive review about a pub visit does not contain information on whether the user was with friends or alone, on a business trip or vacation). We propose to use a small set of manually compiled knowledge resources to predict the associations between the review texts in a user profile and the likely trip contexts. Our experiments, conducted on the TREC contextual suggestion (TREC-CS) 2016 dataset, demonstrate that both factorization and KDE-based generalizations of the relevance model contribute to increased effectiveness of POI recommendation. Further, we demonstrate that trip-qualifier enriched contexts further improve the effectiveness of our proposed model.

As we explore IR-based approaches (specifically pseudo-relevance feedback methods) for contextual recommendation, we also seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents are useful in improving the quality of relevance feedback. To mitigate the problem of over-dependence of a pseudo-relevance feedback algorithm on the top-M document set, we make use of a set of equivalence classes of queries rather than one single query. These query equivalents are automatically constructed either from a) a knowledge base of prior distributions of terms with respect to the given query terms, or b) iteratively generated from a relevance model of term distributions in the absence of such priors. These query variants are then used to estimate the retrievability of each document with the hypothesis that documents that are more likely to be retrieved at top-ranks for a larger number of these query variants are more likely to be effective for relevance feedback. Results of our experiments show that our proposed method is able to achieve substantially better precision at top-ranks (e.g. higher nDCG@5 and P@5 values) for ad-hoc IR and points-of-interest (POI) recommendation tasks. Primary motivation of this part is to achieve better precision at top-ranks by improving the quality of relevance feedback for IR in general, which is eventually applied in the specific task of contextual POI recommendation. POI recommendation, being a precision-oriented task, provides an interesting use-case to study the robustness effects of relevance feedback.

List of Publications

Thesis Related

Parts of this thesis have been published (or under review) in a number of international conferences/journals. Different chapters of this thesis are based on these papers.

- **Anirban Chakraborty**, Debasis Ganguly, Annalina Caputo, and Gareth J. F. Jones. Kernel Density Estimation based Factored Relevance Model for Multi-Contextual Point-of-Interest Recommendation. 2021. In *Information Retrieval Journal*, Springer. (Under review). Preprint available.

  Chapter 5 of this thesis is based on this paper [24].

- **Anirban Chakraborty**, Debasis Ganguly, and Owen Conlan. Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants. 2020. In *29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, Virtual Conference. Pages 125 - 134, ACM, New York, NY, USA.

  Chapter 7 of this thesis is based on this paper [21].

- **Anirban Chakraborty**, Debasis Ganguly, and Owen Conlan. Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation. 2020. In *Proceedings of the 43rd International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '20)*, Virtual Conference. Pages 1981 - 1984, ACM, New York, NY, USA.

  Chapter 6 of this thesis is based on this paper [20].

- **Anirban Chakraborty**, Debasis Ganguly, Annalina Caputo, and Séamus Lawless. A Factored Relevance Model for Contextual Point-of-Interest Recommendation. 2019. In *Proceedings of The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19)*, Santa Clara, CA, USA. Pages 157 - 164, ACM, New York, NY, USA.

  Chapter 4 of this thesis is based on this paper [23].

- **Anirban Chakraborty**. Enhanced Contextual Recommendation using Social Media Data. 2018. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*, Ann Arbor, MI, USA. Pages 1455 - 1455, ACM, New York, NY, USA.

This doctoral consortium paper [18] was drafted based on the initial proposal of this thesis.

Others

A couple of papers [11, 19] have been published as additional work during the period of the PhD.

- Mostafa Bayomi, Annalina Caputo, Matthew Nicholson, **Anirban Chakraborty**, and Séamus Lawless. CoRE: a cold-start resistant and extensible recommender system. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, Limassol, Cyprus. Pages 1679 - 1682, ACM, New York, NY, USA.
- **Anirban Chakraborty**. Exploring Search Behaviour in Microblogs. In *Proceedings of the Seventh BCS-IRSG Symposium on Future Directions in Information Access (FDIA '17).* Barcelona, Spain.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Motivation

Owing to the enormous volume of online data, there is an ever increasing need for contextually relevant recommendations in a variety of domains and use-cases, e.g. recommending movies, articles to read, places to visit, etc. An appropriate definition of *contextual recommendation* obviously relies on a precise definition of the *context* itself. Generally speaking, it can be argued that the more fine-grained the definition of the context is to a system, the better is its potential for providing more personally relevant information to users at specific points in time, specifically focused and tailored to their *context* [5, 96, 67].

To illustrate the point that systems addressing a multiple number of fine-grained contexts are potentially more beneficial to users, imagine two (point-of-interest or POI recommender) systems A and B, where the former only keeps track of a user's geographic location, whereas the latter additionally keeps track of other qualifiers associated with the location, e.g. the specific purpose of the user to visiting that location, whether the user is alone while visiting the place or she is with her friends or family, the season, day or hour of the visit, etc. It can be hypothesized from this example that system B, in comparison to system A, could potentially provide more selective and relevant recommendations to its user about places to visit, and activities to do. This is because system B could potentially reason that museums would not be the best place to recommend if the purpose of the user's current trip is business. On the other hand, it is rather difficult for system A to exclude such non-relevant suggestions because of the lack of adequately informative context.

In addition to the context, the other source of useful information for contextual recommendation is the user's personal *history* or *activity log* [67, 59]. The rationale for using the personal historical information of the user is based on the assumption that user feedback (in the form of ratings or positive/negative comments) may help to capture her preferences. Consider, for example, if the user is particularly fond of live music (i.e., she has in the recent past favoured pubs offering live music over the ones which do not, and has also rated them positively), it is likely that suggesting a pub with live music in a new location could also be relevant to her. Specifically, a contextual recommender system could attempt to *match* a user's past preferences in other contexts (e.g. locations) with the top rated points-of-interests (POIs) of the current context to suggest potentially relevant ones [90].

From a general perspective, we consider that there are two broad distinct sources of information (or contexts), that a contextual recommendation system can benefit from. The first of these describes the *present state* of the user at an instant of time, which is typically a combination of features with categorical values [56], e.g. the location of the user (one out of a finitely many cities on Earth), purpose of a trip (e.g. leisure vs. work), current season (e.g. summer, fall, winter or spring), etc. The second source of information is the *past state* of the user, which, acquired over a sufficient period of time, is likely to broadly capture her general preferences in particular situations. In other words, past information provides information about a user's general preferences for certain types of items over others [67, 59], e.g. 'museums' over 'beaches', e.g. when travelling 'solo' (*accompanied-by* qualifier) for 'leisure' (*trip-type* qualifier).

To illustrate the potential usefulness of both the *present state* and the *past state* contexts with an example, consider the situation when a user visits Dublin with her group of friends in early summer. Based on the user's previous preferences in other locations (e.g. the user usually loves to hangout with friends, or she is an avid draught lover, or she loves trekking or hiking), a context-aware system should seek to *match* information from previous user preferences with the POI descriptors in the current context. For this example, an ideal system should recommend popular tourist destinations and activities in Dublin, that match the user preference history, such as the cliff walk in Howth, the Guinness Storehouse, Temple Bar, etc.

In addition to semantically matching the present state POI descriptors and the past preferences based on the *present state* context of a given user location, an effective system should also consider the more personalized *present state* context qualifiers, such as trip-type, accompanied-by etc. [1]. Again as an example, a user's visit to Dublin for leisure with a group of friends should lead to preferring such suggestions as 'lunch at cheap prices in pubs at the Temple Bar region' over the ones such as 'lunch at the restaurant Avoca', because the latter is more suitable for families.

There are two fundamental differences between the location qualifier and the rest of the context qualifiers. Firstly, the location of a POI is a universal property (irrespective of the perspective of individual users) whereas the other qualifiers, e.g. 'trip type', 'time of travel' etc., are intricately tied as attributes of individual users. Secondly, the location information of a user acts as a *hard* constraint for POI recommendation because for a contextual suggestion to be meaningful and usable, the locations of the recommended POIs must be close to the present state location of the user. On the other hand, non-location qualifiers do not enforce a hard constraint, e.g. a positively rated POI in the past for a trip-type which was different from the current one (e.g. 'solo' in the past vs. 'with family' in the present) could still be recommended.

In our experiments the reason to differentiate between the location (city) and the non-location types is due to the difference in nature of the constraints. The location constraint is *hard*, and the system must make recommendations for the current location (city) only because a POI in a different location is obviously non-relevant. On the other hand, the non-location constraint is a *soft* one i.e. a POI which is usually popular for family dinner may still be relevant to a solo traveller.

One may argue that the location context can also be a soft constraint, where accurate geo-coordinates can be taken into consideration for favouring POIs that are in close proximity of the user's accurate coordinates [105]. However, addressing this is beyond the scope of this thesis and is a potential future work, possibly involving simulated users within the geographical bounding box of a city. In the scope of our work in this thesis, a location context refers to a city, which means that recommending POIs outside the city of the user's current (city) location is considered not to be relevant. This is also consistent with the TREC contextual suggestion (TREC-CS) task definition [43], which we also follow for our evaluation framework.

We emphasize that although our experiments are conducted on the TREC-CS dataset, our proposed model is not tailor-made towards the dataset itself. Our model is essentially based on semantically matching a user profile with POI description text, and is hence able to make contextual recommendations in a general scenario, i.e., with the presence of textual user profiles, POI descriptors and optional ratings. Specifically, we assume that each user profile has a number of POIs that the user visited (either liked or disliked) in the past (say in city X, and Y), and a system needs to recommend POIs in her current city say Z (also taking other non-location type constraints into consideration), that she has not visited before.

1.2 IR or Recommender System (RecSys) approach?

After an introduction to the problem, we now discuss two different threads of work that could, in principle, be applied to address this problem of POI recommendation. The first among these is an information retrieval (IR) [65] based approach and the second is one based on recommendation systems (RecSys or RS) [76]. An IR approach uses an analogy that POIs correspond to documents (that are to be retrieved) and the textual representation of a user's past historical preferences broadly corresponds to a query. On the other hand, an RS approach maps users and items respectively to profiles and POIs.

However, a careful consideration of the RS approach reveals that it is most likely not to be effective in the context of our problem, firstly because of the *lack of sufficient data for training* standard RS approaches [5] in learning the user-item associations (e.g. by factorizing a user-item matrix [36]), and secondly because there may be *no ratings available for the POIs in query locations (contexts)*, which is specifically true for our experimental setup.

We argue that contextual POI recommendation is essentially a personalized IR task, where personalized content matching is important. In fact, Arampatzis and Kalamatianos [5] showed that content-based recommendation approaches perform better for this problem. Following this argument, our proposed approach in this thesis is an IR based content matching one. We hypothesize that it is more suitable to formulate the POI recommendation problem as a *constrained IR* problem, which is characteristically different from the scope of a traditional RecSys approach where the popularity of an item depends only on user ratings (e.g. neural collaborative filtering for movie recommendation [45]), or other contextual features.

1.3 Research Objective

Primary objective of this thesis, broadly speaking, is to *explore IR-based approaches for contextual POI recommendation, with a particular focus to improve precision at top ranks*. Specifically, our proposed approaches are based on a (pseudo) relevance feedback [66] framework. In addition, we make an effort to achieve better precision at top-ranks by improving the quality of relevance feedback for IR in general, which is eventually applied in the specific task of contextual POI recommendation. POI recommendation, being a precision-oriented task [1], provides an interesting use-case to study the robustness effects of relevance feedback.

A number of studies have investigated the problem of contextual recommendation from the point of view of matching the content between the POI (document) representation and the user profile (query) representation. Among these, the studies in [94, 49] combined POI-category and bag-of-words similarities between POIs and user profiles. Generally speaking, as POI-categories, these approaches made use of external information from location-based social networks (LBSNs), such as Foursquare[1] or Yelp[2], to match previous user preferences and POIs in the current location. Note that contextual recommendation systems based on this thread of work mainly rely on *exploiting* the existing preferential knowledge of users from their profiles.

On the other hand, a different thread of work [26, 39] utilizes rating-based collaborative filtering, i.e. information from other users to estimate the popularity of a POI in a local context with the hypothesis that POIs with frequent positive ratings from other users could also be relevant to the current user. In contrast to *exploitation*, this collaborative filtering based thread of work primarily relies on *exploring* the POIs using the current context.

However, there is no systematic investigation on the use of user's preference history, top rated POIs in the current context or both, while predicting the appropriateness of a POI for a user in her current context. We propose a generic IR-based framework for contextual POI recommendation, where we infuse user's preference history in past context (*exploitation*) and information about the top rated POIs for the current context (*exploration*) in a systematic manner.

[1] https://foursquare.com

[2] https://www.yelp.com

1.4 Key Research Challenges

In our work, we approach the problem of contextual recommendation from an Information Retrieval (IR) perspective, where POIs can be considered analogous to documents, and the information in the preference history analogous to a query. The key advantage of this approach is that it is mainly unsupervised or weakly supervised. Unsupervised approaches do not need to rely on training a model with annotated data; instead, to make recommendations they rather try to utilize the inherent semantic associations between latent features of the data itself (e.g. semantically matching the past preferences of users with the POI descriptions in the current context). We now highlight the main research challenges in an IR-based approach to contextual recommendation.

1.4.1 Formulation of Query from User History

First, a major challenge in formulating contextual recommendation from an IR perspective is that, in contrast to the traditional IR setup, there is no notion of an explicitly entered user query. In this case, the query needs to be automatically formulated from the information available in the user profiles, such as pieces of text describing their preferences and dislikes. This query then needs to be effectively *matched* with the information of the POIs (documents) in the current context.

1.4.2 Lack of Non-location type Contextual Information in the User History

The second major challenge is the inevitable absence of explicit annotation of non-location type context (e.g. trip qualifiers, such as 'trip purpose' etc.) in the user preference history [20, 43]. To illustrate this point, consider typical user feedback in a Location Based Social Networks (LBSNs), such as Foursquare or TripAdvisor[3]. This usually comprises a text review and an explicit rating score (from very bad to very good). An important point to note here is that this past information usually does not contain trip qualifier information, i.e. the context in which the POI was visited and rated thereafter. Since a user's perception about a POI can be drastically different in changed circumstances, associating a precise context to a preference is useful to model the subtle dependence between the two, e.g. to model the situations that pubs are great for hanging out with friends only when there are no accompanying children, or hiking in the mountains is great only when it is less likely to rain. While on the one hand including this precise context as a part of the user feedback could provide additional sources of information, on the other, it is highly likely to reduce the number of users prepared to submit feedback due to the additional effort required to enter this information through a more complex interface.

1.4.3 Modeling Relevance for Non-location Contexts in the Present State (Query)

While user preference histories generally lack non-location or *trip qualifier*, such information often forms a part of the present state of the user (i.e. the query). In contrast to the situation of a user being not prepared to enter these details every time as a part of feedback to a system, users in this case are more

[3] `https://tripadvisor.com`

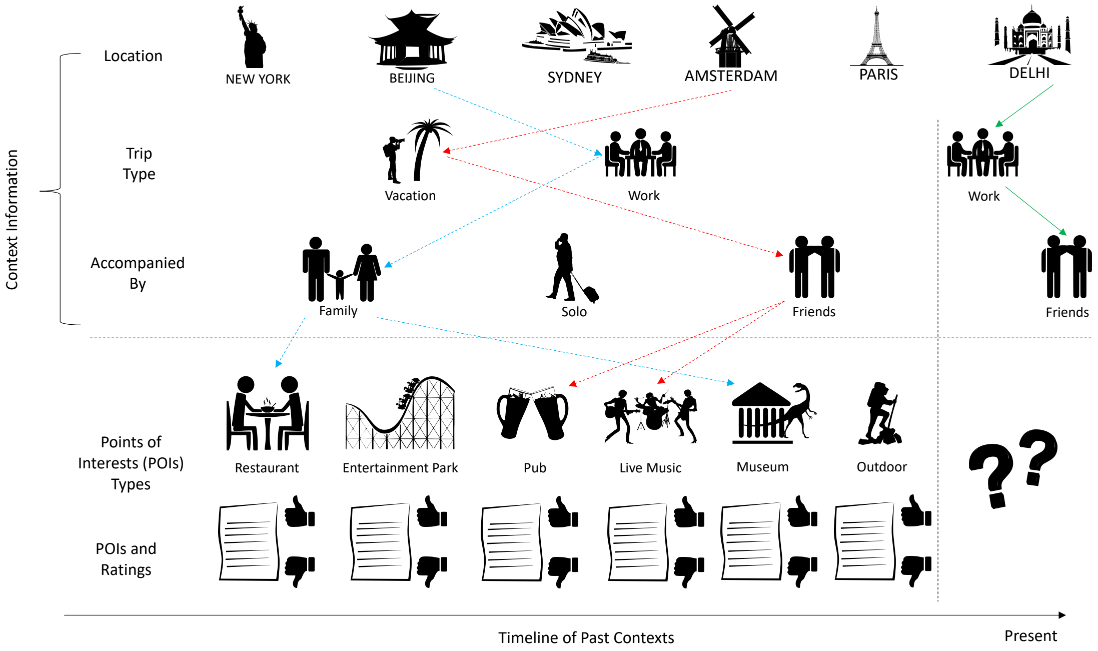


Figure 1.1: Schematic of Contextual Recommendation showing the user's timeline of past and present context(s). Dotted arrows show that the non-location type contextual information (i.e. links between POIs and non-location intermediate nodes) is not present in the user's preference history while both the location and other non-location contexts are available in the present state. We estimate the likely non-location intermediate nodes by utilizing the information from the review text.

likely to submit such information as the type of the trip, whether they are with family or friends etc., because of their intuitive expectation that such precisely defined contextual information (in addition to the current geographic location) would enable the system to suggest more contextually relevant items (POIs). An important research question is then how to bridge the gap between the lack of contextual information from the historical information of user feedback and the constraints imposed by them during the present context (query).

A general approach of bridging this information gap is to employ weak supervision to associate certain topics in user feedback with a seed set of categories defining a precise context, e.g. starting with a seed set of term associations, such as 'pub' being relevant to the context category 'friends'. The natural language text of the reviews is also likely to be helpful in discovering more meaningful dependencies, e.g. associating 'live music' with 'friends', by using the semantic correlation between 'pub' and 'live music'. We propose a formal framework towards this effect.

We illustrate the schematics of the overall idea of the problem and its solution in Figure 1.1. The top part of the figure shows two types of context information of a user, first, the *location* of the user (specifically, a city which the user is currently visiting), and second, the more personal trip-qualifiers (non-location type) information categories which further qualify the location context, e.g. the 'trip purpose' (whether vacation or work), ‘trip type’ (i.e. whether a accompanied by family or a solo trip) etc. The vertical line in Figure 1.1 separates the past context of a user from his present, e.g. the figure shows that the user’s current location is Delhi, and that he has visited New York, Beijing etc. in the past. The bottom-left part of Figure 1.1, constituting a part of a user’s history, shows a list of POIs that the user rated positively (or negatively) during her different trips. We can imagine each unit of context information as a node in a tree that grows downwards from the location nodes to the POI and rating nodes. A path rooted at one of the location nodes and terminating at a particular POI denotes a single trip of a user among her past trips, e.g., in Figure 1.1, the path shown by the red coloured arrows starting from the node ‘Amsterdam’ and visiting in sequence the nodes ‘Vacation’, ‘Friends’, ‘Pub’ and ‘Live Music’ denotes a set of POIs which the user visited (and rated) during her leisure trip to Amsterdam with her friends. Although the complete trip information is shown in the schematic diagram of Figure 1.1, it is worth noting that the tree is essentially incomplete in real-life situation, i.e. the non-location type contextual information is not present in user ratings. The main research challenge is then to estimate a likely path in the tree from a location to a number of POIs, i.e. estimate the likely non-location intermediate nodes by utilizing the information from the review text themselves.

After constructing a model of a user’s preferences, the challenge in contextual recommendation is to be able to make new recommendations to the user for a new present location (that she has not visited before) with a given set of trip qualifiers, e.g., the path specified in Figure 1.1 with the green arrows indicates that the user’s current location is ‘Delhi’ which she is visiting for work along with her colleagues. An effective contextual recommendation system in this scenario should seek to leverage similar situations in the past (i.e. the user’s past non-solo work trips in other locations) in figuring out what type of POIs the user had previously rated positively in those situations, and then use information from these past POIs to recommend a set of similar POIs for the current location.

1.5 Research Questions

Primary research objective of this thesis can be broken into four formal research questions. The first three research questions are directly related to the task of contextual recommendation. We formally define the first research question as,

> **RQ 1:** What is an effective and systematic approach to find the trade-off between a user’s preference history (*exploitation*) and the information about the POIs constrained to a *hard* contextual constraint such as ‘location’ (*exploration*) for contextual POI recommendation?

RQ 1 essentially looks for a systematic way to make a balance between *exploitation* and *exploration* for contextual recommendation, given a *hard* location constraint. The second research question is particularly focused on improving the content matching technique for POI retrieval by incorporating word semantics, and is formalized as,

> **RQ 2:** To what extent, incorporating semantic association between terms present in POI content, while estimating POI's contextual appropriateness, can improve the contextual POI recommendation quality?

The third research question explores a way of incorporating other *soft* contextual constraints, in the above mentioned *hard* constraint based retrieval framework, and is formalized as,

> **RQ 3:** What is the most effective way to include the *soft* contextual constraints such as 'trip-type', 'accompanied-by' of a given user profile into the POI recommendation framework with a particular focus to improve precision at top ranks?

The fourth and the final research question has a wider scope of research contribution. This part is focused on a weakly supervised relevance feedback approach to improve the information retrieval effectiveness in general, which is eventually applied in the specific task of contextual POI recommendation. This is mainly because we are exploring IR-based approaches for contextual POI recommendation, and the primary objective of this part of work is to improve precision at top ranks for IR methods, which is exactly what we are trying to achieve for POI recommendation. In particular, we seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents are useful for relevance feedback. Fourth research question is formalized as,

> **RQ 4:** To what extent retrievability based document selection for relevance feedback can improve the retrieval effectiveness, specifically with respect to precision at top ranks, both in the general ad-hoc IR setup, and for the specific task of contextual POI recommendation?

We address these four research questions through different chapters of this thesis.

1.6 Research Contributions

In this section, we mention the four important research contributions of this thesis, which address the corresponding research questions mentioned in Section 1.5.

1.6.1 Factored Relevance Model (FRLM) to Balance *Exploitation - Exploration*

Before modeling multiple contextual constraints, we first consider the *hard* location constraint only POI retrieval scenario. We propose a formal IR-based approach (specifically, a pseudo-relevance feedback model) to address the problem of contextual recommendation. More specifically, to tackle the problem of matching the user preferences with the POI descriptors in a given query context (essentially the location), we propose a generalized version of the well-known relevance model (RLM) [51]. Our proposed model is a *factored* version of the standard relevance model, where the first step (*exploitation*) involves enriching the user preference information, and the second step (*exploration*) involves subsequently using the enriched information to effectively match the POI descriptors given query context. This part of the thesis finds a solution to the research question **RQ 1**.

We show that the systematic infusion of *exploitation* and *exploration* improves the effectiveness of POI retrieval. This part of work leads to a publication in ACM SIGIR ICTIR 2019 [23]. A characteristic of our proposed relevance feedback based model is that it achieves a sweet-spot between the user's preference history in past contexts (*exploitation*), and the relevance of top-retrieved POIs in the user's current context (*exploration*). Our experiments on the TREC-CS 2016 [43] dataset show that our proposed model of a factored relevance model is able to effectively combine these two sources of information, leading to significant improvements in contextual recommendation quality.

1.6.2 FRLM with Word Semantics for Better Content Matching

Although our experiments (Chapter 4) show that the proposed factored relevance model (FRLM) is effective in matching the content between the POIs in user's preference history and the POIs in the current context by estimating a term weight distribution from both information sources, we have noticed that there exist relevant terms that are not captured by the co-occurrence statistics used in RLM estimation. In particular, improvements in the effectiveness of FRLM [23] was not significant specifically with respect to precision at top ranks (nDCG@5, P@5).

Hence to further improve the retrieval effectiveness at top ranks, we incorporate term semantic information into the FRLM in the form of word vector similarities, and propose a word embedding based (estimated with kernel density estimation) further generalized version of factored relevance model, KDE-FRLM (Chapter 5). This leads to a better semantic match between the POI descriptions and the review/description text of the locations visited in the past by a user, which eventually achieves significantly better retrieval performance. This part of the thesis enlightens readers on the research question **RQ 2**.

Our experimental results show that incorporating word semantics further improve the retrieval performance. Detailed comparative analysis between both the initial FRLM, and the word semantics enriched FRLM reveals that the latter estimates a better term weight distribution for content matching. This part of work, along with the multi-contextual generalization (which we will see next) is currently under review in the Information Retrieval Journal [24].

1.6.3 Multi-Contextual Generalization of FRLM

Initial version of the proposed FRLM addresses the (*hard*) location context only, and ignores other non-location type qualifiers. We further generalize the proposed initial framework by introducing multiple contextual constraints. This part contributes to two factors.

Firstly, we incorporate a *generalized framework of addressing both the hard and the soft constraints* (location and trip qualifiers respectively) within the framework of the proposed relevance model. We undertake a weakly supervised approach (leveraging a small set of context-term annotations) to transform the *soft* constraints into term weighting functions.

Further, we incorporate *term semantic information* within the framework of our proposed relevance model. In particular, we use embedded vector representations of words to bridge the vocabulary gap between user preferences, POI descriptions and the trip qualifier (*soft*) constraints.

This part of the thesis addresses the third research question, **RQ 3**. Our experiments show that the weakly supervised approach of modeling multiple *soft* constraints further improves the POI recommendation quality. This multi-contextual generalization of the FRLM leads to a publication in ACM SIGIR 2020 [20].

We would like to mention that the word embedding based factored relevance model (KDEFRLM) has been developed for both the location only (i.e. *hard* context based) retrieval, and the multi-contextual (i.e. *hard+soft*) recommendation. The inclusion of word embedding within the framework (i.e., KDEFRLM) is able to achieve significant improvements over a number of IR-based, and RecSys-based baselines.

In addition, we also investigate the choice of *different word embedding techniques (in-domain vs. externally trained)* in the effectiveness obtained with our proposed model KDEFRLM (in both the kernel density estimation process and also in modeling the *soft* contextual constraints).

1.6.4 Retrievability based Document Selection to Improve Precision at Top Ranks

As mentioned earlier, this part of work particularly explores a way of improving precision at top ranks for IR methods (specifically pseudo-relevance feedback methods) by selecting useful feedback documents. We propose a concept of weakly supervised relevance models by using the notion of *retrievability* [6] from automatically constructed query variants to improve the quality of relevance feedback. This part of work is related to the final research question, **RQ 4**.

We observe that our approach consistently improves precision at top ranks in two different tasks, namely TREC ad-hoc and the contextual POI recommendation. This work leads to a publication in ACM CIKM 2020 [21].

1.7 Thesis Outline

The rest of the thesis is organized as follows.

Chapter 2 revisits some standard background about IR including the concept of retrieval models, relevance feedback, and IR evaluation methodology. We also survey existing work on contextual recommendation including different IR-based and RecSys based approaches, that are related to our work.

Chapter 3 provides the details about the proposed IR-setup, evaluation framework, and the data sets used, which are common for all subsequent experiments. To illustrate some of the basic concepts of our proposed models (i.e. matching the documents and the queries), we have shown real examples from the TREC-CS dataset. We include an example user profile, and a document (POI) representation. This also explains why TREC-CS is a suitable dataset for our experiments

Chapter 4 presents the proposed novel factored relevance model (FRLM) for contextual POI recommendation, particularly to tackle the problem of matching the user preferences with the POI descriptors in a given location (*hard*) context. We compare the performance of the FRLM with a number of standard IR-based and RecSys based approaches in the same experimental setup.

Chapter 5 generalizes the proposed FRLM to include word semantic information for a better content matching between the POIs in the past contexts and the POIs in the current context. We present a comparative analysis between the initial FRLM, and the word semantics enriched FRLM, while addressing only the location constraint. We show that in addition to the factorization, word semantics based generalizations of the relevance model contribute to increased effectiveness of POI recommendation.

Chapter 6 further generalizes FRLM to the multi-contextual case by incorporating term preference weights corresponding to trip qualifier (*soft*) constraints. We provide details about both the initial FRLM, and the word semantics enriched FRLM, while addressing both the *hard*, and other *soft* contextual constraints. We demonstrate that trip-qualifier enriched contexts further improve the effectiveness of our proposed models.

Chapter 7 proposes a concept of weakly supervised relevance models by using the notion of retrievability from automatically constructed query variants to improve the quality of relevance feedback, with a particular focus to improve precision at top ranks in two tasks, namely ad-hoc IR, and contextual POI recommendation.

Chapter 8 enlists primary contributions, key findings, achievements, and discusses how the research questions have been addressed through different chapters, and eventually concludes the thesis with directions for future work.

Chapter 2

Background

We already mentioned that contextual (POI) recommendation is essentially a personalized Information Retrieval (IR) task, and we particularly explore relevance feedback based approaches for this task. In this chapter, we first briefly examine the background of IR including the concept of retrieval models, and relevance feedback. We also discuss how the effectiveness of an IR system can be evaluated. Finally, we survey existing work on contextual recommendation.

2.1 Information Retrieval

Retrieval from web is perhaps the best known example of Information Retrieval (IR), where a search system returns information (usually a set of documents, images, videos etc.) that are relevant to a user's information need. In this thesis, we particularly focus on *textual information*, which is in fact one of the most prevalent forms of information even in today's world.

Information retrieval process, broadly speaking, consists of two components: 1) *indexing*, and 2) *retrieval*. To achieve fast and efficient retrieval, the set of documents, which is commonly known as a corpus, is processed and stored by making use of an *inverted index* during indexing. In this thesis we make use of *static indexing*. *Dynamic indexing* may be useful for commercials search engines, where the corpus may not be static, and the existing index may need to be updated dynamically with new documents without creating a new index from the beginning. However, it is a common practice to make use of a static collection for laboratory based experiments, specifically for research purposes.

All *non-informative* terms such as *stopwords* are removed before indexing. Conceptually inverted index is comprised of an *inverted list* or *posting list*, which stores each vocabulary term along with a list of documents containing that term, and a *dictionary*, which stores all the terms in the vocabulary. The inverted index usually contains document specific term weights, and the dictionary may contain term specific collection statistics. This makes it easier to get all document specific necessary information to compute the retrieval score of a document.

2.1.1 Retrieval Models

Retrieval models or retrieval functions are the ones that work behind a search system to retrieve information that are relevant to a user query (representation of the actual information need [28]). A number of algorithms / functions are known to be effective in the research community. In this section, we will describe a couple of well known models that are eventually employed in this thesis.

The use of language model based retrieval methods, particularly with Dirichlet smoothing and Jelinek Mercer smoothing [102, 103, 104] is widespread. These approaches are frequently employed by the researchers as baselines [99, 106, 40, 73, 34, 22, 37, 38]. In our work, we also experimented with a probabilistic retrieval model, specifically BM25 [77, 81], which is in fact considered to be a strong baseline in IR experiments [57]. Now, we briefly discuss about these models, and then we explain relevance feedback, and query expansion methods, which have been applied in our work.

Language Model

Language modeling based retrieval is primarily motivated by the *probability ranking principle* [79, 46]. Let D be the language model estimated from a document d. To compute the score of a document d for a given query Q, posterior probability $P(D|Q)$ are estimated for each document d in the collection by making use of the prior probability $P(Q|d)$ based on the Bayes rule [75, 46, 104].

$$\begin{aligned} P(d|Q) &= \frac{P(Q|D) \cdot P(D)}{\sum_{d' \in C} P(Q|D') \cdot P(D')} \propto P(Q|D) \cdot P(D) = P(D) \cdot \prod_{q \in Q} P(q|D) \\ &\propto \prod_{q \in Q} P(q|D) \end{aligned} \tag{2.1}$$

Th language model D of the document d is approximated usually by unigram model, $D = \{P(w_i|d)\}$, where $i \in [1, |V|]$, V being the vocabulary. $P(w_i|d)$ is the probability of sampling the term w_i from the document d. Hence, the retrieval score of the document d, with respect to the given query Q can be represented as,

$$\begin{aligned} S(Q, d) &= P(Q|D) \\ &= \prod_{q \in Q} P(q|d) \end{aligned} \tag{2.2}$$

Jelinek Mercer Smoothing An inevitable problem with Equation 2.2 is that the document score can become zero when a query term is not present in the document d. To overcome this problem, the language model D can be smoothed by interpolating a language model estimated from the whole collection C, with the Maximum Likelihood Estimate (mle) of $P(w_i|d)$ as,

$$\begin{aligned} P(Q|D) &= \prod_{q \in Q} [\lambda P(q|d) + (1 - \lambda) P(q|C)] \\ &= \prod_{q \in Q} \lambda \frac{tf(q, d)}{|d|} + (1 - \lambda) \frac{cf(q)}{|C|}), \end{aligned} \tag{2.3}$$

where, $tf(q,d)$ is the term frequency, i.e. the number of occurrences of term q in the document d, and $|d|$ is the size of the document. Similarly, $cf(q)$ is the collection frequency of q, and $|C|$ is the size of the collection. Interpolation parameter $\lambda = [0,1]$. The retrieval model based on this language modeling technique (Equation 2.3) is known as language model with linear smoothing or Jelinek Mercer smoothing.

Dirichlet Smoothing Dirichlet smoothing is another popular smoothing technique, which makes use of Bayesian estimation instead of maximum likelihood estimation as,

$$P(Q|D) = \prod_{q \in Q} \frac{tf(q,d) + \mu P(q|C)}{|d| + \mu}, \tag{2.4}$$

where, $tf(q,d)$ is the term frequency, i.e. the number of occurrences of term q in the document d, and $|d|$ is the length of the document. $P(q|C)$ is the collection probability of the query term q in the entire collection. Here, the interpolation parameter μ (usually set in the range $[100, 5000]$) has a dynamic coefficient that changes based on the document length. The retrieval model based on this language modeling technique (Equation 2.4) is known as language model with Dirichlet prior smoothing.

BM25

BM25 is a traditional probabilistic retrieval model [80, 50, 77, 81] which is based on the *probability ranking principle* [79, 46], and essentially estimates the posterior probability of a document d, being relevant to a query Q. Score of a document d with respect to the query Q can be computed as,

$$S(Q,d) = \sum_{q \in Q} \log \frac{N - df(q) + 0.5}{df(q) + 0.5} \cdot \frac{tf(q,d)(k_1 + 1)}{tf(q,d) + k_1(1 - b + b\frac{|d|}{avgdl})}, \tag{2.5}$$

where, N is the number of documents in the collection, $df(q)$ is the document frequency i.e. the number of documents (in the entire collection) in which the term q occurs, $tf(q,d)$ is the term frequency, i.e. the number of occurrences of term q in the document d, $|d|$ is the document length, and $avgdl$ is the average document length of the whole collection. Parameter k_1 calibrates the term frequency (tf) contribution [81]. While a higher value of k_1 favours the contribution of the tf factor, a lower value of k_1 decreases the importance of the tf factor. On the other hand, parameter b controls the length normalization factor. Indeed BM25 is a strong baseline retrieval model with better length normalization factors, which we have employed in all our experiments.

2.1.2 Relevance Feedback

A prevalent problem of the retrieval models is the problem of *vocabulary mismatch* [33], which is in fact a common IR challenge. Consider a document D containing a term/phrase ‘nuclear power’ is relevant to a query Q containing a term/phrase ‘atomic energy’. Despite referring to the similar concept, classic retrieval models may fail to retrieve the document D, in response to the query Q due to vocabulary mismatch (i.e. use of different set of words by Q, and D).

Overall idea of *relevance feedback* in IR is to involve users in the retrieval process by incorporating their feedback, to improve the final retrieval performance. Particularly, after initial querying users are provided with the initial set of retrieved documents. Users provide their feedback on the relevance of these documents. The retrieval system then modifies the initial information need based on this feedback, usually by expanding the initial query [78, 51], and performs a second step retrieval.

Generally speaking, there are three types of relevance feedback, based on the way the feedback is obtained.

Explicit Relevance Feedback User feedback is obtained by asking users to explicitly provide feedback on the relevance of the initial set of retrieved documents. In practice, users are not always interested to provide explicit feedback.

Implicit Relevance Feedback User feedback is obtained based on user behaviour such as previous search history. Usually users are not aware that their search behaviour may be used to better understand their interests.

Pseudo Relevance Feedback After querying with the initial query, the set of top retrieved documents are considered relevant.

In reality, while explicit and implicit feedback are hard to get, pseudo relevance feedback methods turn out to be effective [86, 51]. In this section, we will explain pseudo relevance feedback based query expansion technique. In particular, we will discuss about relevance model (RLM or RM for short) [51], which is a popular and effective query expansion technique.

The relevance model (RLM) [51] is a (pseudo) relevance feedback (PRF) method which estimates a term's importance for relevance feedback by using the co-occurrence information between a set of given query terms and those occurring in the top-retrieved documents. RLM hypothesizes that the terms frequently co-occurring with a query term are semantically related to the information need and, therefore, could be used to enrich the query with additional information.

Formally speaking, given a query $Q = \{q_1, \ldots, q_n\}$, the RLM estimates a term weight distribution from a latent relevance model R, $P(w|R) \approx P(w|Q)$, from a set of M top-retrieved documents $\mathcal{M} = \{D_1, \ldots, D_M\}$, as shown in Equation 2.6.

$$P(w|R) \approx P(w|Q) = \sum_{D \in \mathcal{M}} P(w|D) \prod_{q \in Q} P(q|D) \tag{2.6}$$

From Equation 2.6, it can be seen that higher $P(w|Q)$ values (RLM term weights) are obtained for a term w if it occurs frequently in a top-ranked document (large $P(w|D)$ value), in conjunction with the frequent occurrence of a query term, i.e. a term $q \in Q$ such that $P(q|D)$ is also large.

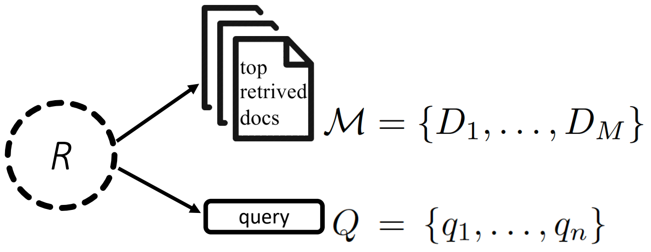

R
top retrived docs
$\mathcal{M} = \{D_1, \ldots, D_M\}$
query
$Q = \{q_1, \ldots, q_n\}$

Figure 2.1: Both the query Q and its relevant documents (where the set of top retrieved documents, $\mathcal{M}$ are considered relevant) are sampled from a latent relevance model R (Lavrenko and Croft [51]).

In other words, RLM assumes that there is a latent relevance model R, and both the query and its relevant documents are sampled from it, which is getting estimated (as shown in Figure 2.1). This joint probability estimation of $P(w|Q)$ is basically *independent and identically distribute (i.i.d.)* sampling. Another possible approach for this probability estimation is the conditional sampling.

While the original version of the relevance model (i.i.d. sampling) is commonly known as 'RM1' in the literature [62], its conditional sampling version is known as 'RM2'. 'RM1' does not take the original query terms into account while estimating the density function, which usually results in a query drift [62]. It has been shown that a mixture model of the estimated density of other term weights in conjunction with the original query terms yields more robust feedback results [62]. This mixture model, commonly known by the name 'RM3' [48], is represented as shown in Equation 2.7.

$$P'(w|R) = \lambda P(w|R) + (1 - \lambda)P(w|Q) \tag{2.7}$$

Each mention of 'relevance model' or 'RLM' or 'RM' in this thesis is to be interpreted as its more effective mixture model variant, i.e. 'RM3'.

2.1.3 Evaluation Methodology

Cranfield tradition research paradigm was introduced in the 1960s [27] which shaped the IR evaluation. Following this paradigm, several standard test collections were prepared by TREC[1] including TREC ad-hoc and TREC contextual suggestion (TREC-CS), where a collection is composed of three primary components: i) a static corpus, i.e. a set of documents, ii) a set of queries, and iii) relevance judgements. Effectiveness of an IR system can be empirically validated in this evaluation framework.

We have used the official *trec_eval*[2] evaluation tool for evaluating the performance of all methods employed in this thesis. *trec_eval* is a standard tool widely used by the IR research community for evaluating a retrieval run, given a set of relevance judgements, and the results file generated by the IR system. *trec_eval* reports a number of standard evaluation metrics such as mean average precision (MAP), and

[1] https://trec.nist.gov/

[2] https://trec.nist.gov/trec_eval/

Collection	#documents	Topic Set	#Topics	Fields	Qry Ids	Avg. $\lvert Q \rvert$	Avg. #Rel
Disks 4 and 5 minus CR	528,155	TREC 6	50	title	301-350	2.48	92.22
		TREC 7	50	title	351-400	2.42	93.48
		TREC 8	50	title	401-450	2.38	94.56
		TREC Rb	99	title	601-700	2.88	37.20
TREC-CS 2016	1.2 M	Phase 1	61	tags	700-922	10.36	35.26

Table 2.1: Overview of datasets (TREC ad-hoc, and TREC contextual suggestion) used for different experiments mentioned in this thesis.

precision at a specific rank cut-off ($P@k$) for binary relevance, or normalized discounted cumulative gain ($nDCG$), in case of graded relevance, etc. We will discuss about these evaluation metrics later in this section.

Datasets

Particularly, for contextual recommendation experiments (detailed in Chapter 3 - 6), we have used TREC contextual suggestion (TREC-CS) 2016 collection [43]. For our experiments on relevance feedback with query variants (Chapter 7) to improve retrieval effectiveness for ad-hoc retrieval, and contextual recommendation task, we have used TREC 6 - 8 [42, 92] and robust [93], and TREC-CS collection, respectively. Table 2.1 provides an overview of datasets used in our experiments.

The primary reason behind selecting TREC-CS for our contextual recommendation experiments is that it provides a controlled evaluation framework for researchers working on the contextual recommendation problem, with well defined (pool based) relevance judgements, which is a key component in *Cranfield tradition* research paradigm [27]. To illustrate some of the basic concepts of our proposed model (i.e. matching the documents and the queries), we have shown real examples from the TREC-CS dataset. We include an example user profile, and a document (POI) representation. This also explains why TREC-CS is a suitable dataset for our experiments. Details are to be found in Chapter 3 (Section 3.2.2).

On the other hand, as TREC Robust collections are known as good collections for evaluating pseudo-relevance feedback, and widely used by researchers [42, 92, 93, 51, 84] worked in this area, we also use this data for our experiments on relevance feedback with query variants.

Evaluation Metrics

Here, we discuss about a couple of well known evaluation metrics that are used to evaluate the effectiveness of our proposed approaches, and other baseline approaches. Let $\mathcal{Q}$ be a set of queries corresponding to the set of actual information needs, and Rel_Q be the number of documents that are known to be relevant to a query $Q \in \mathcal{Q}$. Now we would like to evaluate the effectiveness of a retrieval function ϕ which retrieves a set of documents $\mathcal{L} = \{d_1, d_2, \ldots, d_n\}$ ranked based on their retrieval scores $S(Q, d)$, where $S(Q, d_i) \geq S(Q, d_j), \forall i < j$.

Mean Average Precision (MAP) Among other evaluation metrics, Mean Average Precision (MAP), which is a single-value metric to measure retrieval effectiveness across recall levels, has satisfactory discriminative property with notable stability [65].

$Rank(d_i) = i$ i.e., the rank of each document d_i, where $i = 1, 2, \ldots, n$ is defined to be the position in the rank list at which d_i is retrieved. For each query Q, let $RelRet_Q = \{d_1, d_2, \ldots, d_m\}$ be the m relevant documents ($m \leq n$) that are retrieved among Ret_Q documents ($Ret_Q \subseteq RelRet_Q$). The average precision (AP) for the query Q can be computed as,

$$AP(Q) = \frac{1}{|Rel_Q|} \sum_{d_m \in RelRet_Q} \frac{m}{Rank(d_m)} \tag{2.8}$$

Then the mean average precision (MAP) is computed by averaging $AP(Q)$ values over the set of queries $\mathcal{Q}$ as,

$$MAP(\mathcal{Q}) = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} AP(Q) \tag{2.9}$$

The metric is calculated based on the top n retrieved documents, where n is usually set to 1000. While we have set $n = 1000$ for the TREC ad-hoc task (Chapter 7), n is set to 50 for the task of contextual recommendation (Chapter 3 - 6) as instructed in the TREC contextual suggestion task description [43].

Precision at k ($P@k$) Precision at rank k ($P@k$) can be computed as,

$$P@k = \frac{Rel(\mathcal{L}_k)}{k} \tag{2.10}$$

where, $Rel(\mathcal{L}_k)$ is the total number of relevant documents present in the rank list $\mathcal{L}_k$, up to rank k. Similar to MAP, $P@k$ is computed by averaging $P@k$ values over the set of queries $\mathcal{Q}$.

2.2 Contextual Recommendation

Broadly speaking, there are two widely known approaches in the literature of recommender system research: (i) Content-based filtering, and (ii) Collaborative filtering. For a traditional (POI) recommender system, content-based filtering is essentially a content matching problem between a candidate POI description and the POI descriptions in user's preference history, assuming that the user would like similar POIs as she liked in the past. On the other hand, collaborative filtering utilizes information from other similar users to estimate the relevance of a POI relying on the hypothesis that POIs with frequent positive ratings from other similar users could also be relevant to the current user.

In addition, a context-aware POI recommender system should consider the appropriateness of a POI in the current contexts. For example, even if a user's preference history indicates that the user is an avid beer lover, it may not be suitable to suggest pubs to this user when she is out with her family in the morning. It is important to point out that the context can have multiple dimensions: temporal (such as ‘day’, ‘night’, ‘summer’, ‘winter’ etc.), geographical (such as the current ‘city’ or a precise ‘location’ in the city), personal (such as age or gender), trip type (such as ‘with family’ or ‘business trip’) etc.

2.2.1 Content-based Exploitation

The problem of contextual recommendation has been investigated by a number of studies from the point of view of matching the contents of a user profile (query) representation and the POI (document) representation. Among these, the studies in [94, 49] combined similarities between POI categories and user profile content. Generally speaking, for the POI categories, these approaches made use of external tag information from location-based social networks (LBSNs), such as Yelp or Foursquare, to match past user preferences and POIs in the current context.

The contextual suggestion track[3] (TREC-CS) provides a common evaluation platform for researchers working on the contextual recommendation problem. Given a set of example POIs which reflect the user’s past preferences, and some contextual information such as temporal, geographical and personal contexts, the task was to return a ranked list of POIs that fits the user profile and current context. The task tests if a system can recommend POIs effectively in a new city, say New York, when the system has the previous knowledge of user’s preferences in other cities, such as Seattle or Detroit. A very popular approach among the task participants was to retrieve POIs from different LBSNs such as Google Place, Foursquare or Yelp based on geographical context, and then to apply some heuristics such as “night club will not be preferred in morning” or “museum will be closed at night” to filter out POIs that do not match the given temporal context [29, 43]. Arampatzis and Kalamatianos [5] experimented with different content-based, collaborative filtering based and hybrid approaches on TREC-CS, and found that the content-based approaches performed better than other approaches.

Most of the TREC-CS participants formulated the task as a content-based recommendation problem [94, 49, 82, 55]. A common approach was to estimate a user profile based on the POIs that the user preferred previously, and then rank the candidate POIs based on their similarities to the estimated profile, assuming that a user would prefer POIs that are similar to those they liked before. Some of these studies used the descriptive information of the POIs and/or the web pages of the preferred POIs to build user profiles, and then used several similarity measures to rank the POIs [94, 49].

The authors of [54, 55] explored the use of LBSNs’ category information for user modeling and POI ranking. In particular, after gathering available user profile information, Li et al. [54] modeled user profiles and employed a binary classification (for user’s *like*, and *dislike*) to classify POIs, based on their categories such as restaurant, shopping, nightlife etc. Li and Alonso [55] proposed a Reinforcement and Aging Modeling Algorithm (RAMA) to construct user profiles, where each user profile had two components: i) general interest model which was comprised of Yelp category information such as gallery,

[3] `https://sites.google.com/site/treccontext/`

museum, landmark etc., and ii) specific interest model which was comprised of content words. For ranking, the score of a candidate POI was computed based on the current context and the combined similarity with both the general, and specific user model.

A recent work by Aliannejadi and Crestani [2], then extended in [1], applied linear interpolation and learning-to-rank to combine multiple scores such as review-based score and tag matching score for context-aware venue suggestion. The motivation behind using a review-based score was to better understand the user's motivation behind rating a POI (liked, or disliked). They trained a binary SVM classifier by considering review texts from positively rated POIs as positive samples, and review texts from negatively rated POIs as negative samples. On the other hand, tag matching score contributed to a similarity measure between POIs by making use of Foursquare, and Yelp category tags.

It is becoming increasingly popular among researchers to make use of online user reviews in different ways for contextual recommendation, such as by learning the importance of user ratings, by learning the latent topic, or contexts present in the review text [25]. Musat et al. [71] made use of weighted ratings. Specifically they consider the topics mentioned in both the candidate POI's review text, and the review text present in the user profile. The similarity between these topics were then used for ranking.

Use of a single LBSN may not be sufficient to capture the information about all POIs and/or all the available types of information about the POIs. In another recent study, Aliannejadi et al. [4] show that the amalgamated use of a user's current context and the ratings and reviews of previously rated POIs from multiple LBSNs improve recommendation accuracy. They crawled both Foursquare, and Yelp to acquire additional information about POIs such as category, keywords, review text etc. This thread of work for contextual recommendation is mainly based on *exploiting* the user's existing preference history information and essentially performs content matching between the POIs in the user's preference history and the candidate POIs.

2.2.2 Collaborative Filtering based Exploration

A different thread of work [26, 39] makes use of rating-based collaborative filtering, i.e. information from other users, to estimate a POI's popularity in a current context with the hypothesis that POIs with frequent positive ratings from other users could also be appropriate to the current user.

Recommendation-based algorithms mainly involve applying rating-based collaborative filtering approaches that are based on finding features that are common among multiple users' interests, and then recommending POIs to users who share similar preferences. Matrix factorization, a standard technique that represents both users and items in a latent space, forms the core of most of these recommendation based approaches. It is common to make use of the check-in information collected from LBSNs for recommending POIs [26, 39]. Cheng et al. [26] employed a multi-center Gaussian model to model the geographical influence by estimating the probability of a user's check-in on a POI location, and incorporated this into a generalized matrix factorization framework along with a social influence model [64, 63]. On the other hand, Griesner et al. [39] integrated both the geographical influence, and temporal influences of POIs' check-in, into a matrix factorization model.

However, collaborative filtering based techniques often suffer from the data sparsity problem. This problem is even worse for POI recommendation where a single user can only visit (and rate) a small number of the POIs available in a city. As a result, the user-item matrix [36] becomes very sparse [96] which leads to poor recommender system performance. Due to this data sparsity problem, it can be difficult for purely recommendation based approaches to yield effective outcomes for POI recommendation.

Some existing work [95, 97] has addressed this data sparsity problem of collaborative filtering by incorporating supplemental information into the model. Specifically, Ye et al. [95] argued that the spatial influence of locations affects users' check-in behaviour. They incorporated spatial and social influence to build a unified location recommender system. On the other hand, the system developed by Yuan et al. [97] which is a time-aware collaborative filtering model, recommends locations to users at a certain time of the day by leveraging other users' historical check-in information. They argued that people show a periodic behavior throughout the day. Hence they split a day into a number of time slots, and modeled the user check-in behaviour in different time slots of the day. Eventually they introduced time as an additional dimension into a standard user-item matrix.

To address the cold-start situation for hotel recommendation, Levi et al. [53] designed a context-aware recommender system. They constructed context groups based on user reviews and regarded the user's preferences in trip intent i.e. the purpose of the trip, and the similarity of the current user with other users such as their nationality. They also consider user preferences for different hotel features in their model. Fang et al. [32] developed a model that consider use of both spatial and temporal context information to handle the data sparsity problem. Their model STCAPLRS is comprised of two major components i.e. offline user modeling and online recommendation. They designed a regression mixture model to learn individual user preference and the local preference of a specific location. Online component recommends POIs based on the user model, the current spatial-temporal context, local preference etc.

Existing research that use time as a context includes [35, 30]. Deveaud et al. [30] designed a time-aware venue suggestion system which modeled popularity or appropriateness of venues (POIs) in the immediate future with the help of time series. In contrast to *exploitation*, this thread of work for contextual recommendation primarily relies on *exploring* the candidate POIs using the current contextual information.

Chapter 3

Research Framework

In this chapter, we formally define a standard IR research framework for contextual recommendation and provide details about data sets used. We then explain the experimental setup where we perform different experiments with our proposed model(s) and a number of comparative baseline approaches. This experimental setup and data sets are in fact common for all subsequent experiments.

3.1 IR Setup Foundation

Unlike the traditional IR setup, there is no explicit user query in contextual recommendation (CR). The primary objective in CR is rather to match the user's preference history with the POI descriptors (analogous to documents) in the user's current context(s). This contrasts with an IR-based approach where an explicit query can be formed from bits of information from the user profile.

3.1.1 Notations for User Profile

A user profile is comprised of a descriptive text, a set of tag terms added to it and a score (see the bottom part of Figure 1.1). It should be noted that a document representation in a user profile does not have information about trip qualifiers, as indicated by the dotted arrows from the upper part of Figure 1.1 into each review (mentioned in Section 1.4.2). The current context of a user forms a part of the query comprised of a pair of trip qualifiers of the form (L, Q), where L is the location (*hard*) context, and $Q = Q_1 \times \ldots Q_c$ is a combination of c non-location (*soft*) contexts. The general definition allows c to be any finite integer. As per our experiments with the TREC-CS 2016 dataset [43] the available number of such non-location qualifiers is $c = 3$, i.e. the value of c specifically for our experiments is 3. In particular, Q_1=`trip-type`, e.g. vacation, Q_2=`trip-duration`, e.g. day-trip, and Q_3=`accompanied-by`, e.g. solo or with friends. Each non-location type context q_U is hence a 3-dimensional categorical vector.

Indeed, in a general case it should be possible to include a number of contextual constraints such as geographical influence [95], time of the day [97], road traffic or availability of transportation, current

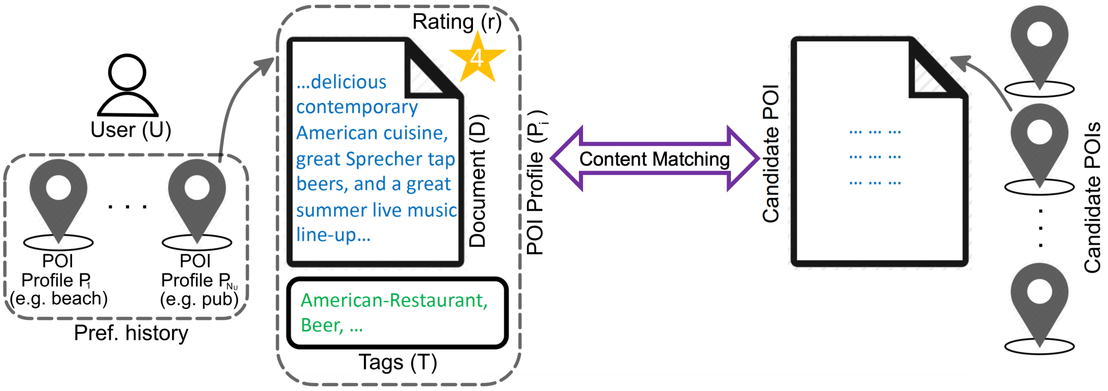

User (U)
POI Profile P1 (e.g. beach)
. . .
POI Profile PNU (e.g. pub)
Pref. history
Rating (r)
4
...delicious contemporary American cuisine, great Sprecher tap beers, and a great summer live music line-up...
Document (D)
POI Profile (Pi)
American-Restaurant, Beer, ...
Tags (T)
Content Matching
Candidate POI
... ...
... ...
... ...
Candidate POIs

Figure 3.1: Pictorial depiction of the IR setup for contextual suggestion, which essentially involves matching the content between a candidate document to be retrieved (i.e. a POI description) and a textual representation of a user profile of the form $P_i = (D, T, r) \in U$ in the user's preference history.

weather etc. as a part of the non-location type constraints (i.e. use a value of c higher than that of 3). However, we restrict the scope of our investigation to three specific non-location type attributes only and leave the other attributes for a possible future extension of this work.

From a general IR point-of-view, we assume that a user profile U is composed of a set of N_U profile P_i's and an instance of the user's current context specified by the location and trip qualifiers $(l_U, q_U) \in (L, Q)$. Each profile P_i is a 3-tuple consisting of a document (D which belongs to a static collection $\mathcal{D}$), a set of user assigned tags (T which is a subset of a controlled tag vocabulary $\mathcal{T}$), and a user provided rating (r normalized within $[0, 1]$, higher the better). This is stated formally in Equation 3.1.

$$U = \cup_{i=1}^{N_U} \{P_i : P_i = (D, T, r) \in \mathcal{D} \times \mathcal{T} \times [0, 1]\} \tag{3.1}$$

The objective of a tag $t \in \mathcal{T}$ is to express a POI as a set of single words or short phrases that best represents the POI, real instances of which are 'beer', 'American Restaurant', etc. assigned to the POI e.g. a restaurant. The document representation of the POI is composed of the text description accumulated from the POI's home page, customers' reviews on social networks etc. The definition of every each document in the collection is assumed static.

For the sake of convenience in referring back to the notations, we give their definitions in Table 3.1.

3.1.2 Retrieval with the Location Constraint

The objective then is to *rank* a set of POIs (hard constrained by $L = l_U$) in decreasing order of their estimated relevance scores within the current context. A simple way to estimate the relevance scores is to first restrict the set of candidate POIs to only the ones in the specific location (by employing the hard constraint), i.e. $S(l_U) = \cup\{d : L(d) = l_U\}$ (L denoting the location attribute of a POI). The next step then makes use of the text in the user profile, U, and this candidate set of POI descriptors $S(l_U)$ to

Notation	Implication
$\mathcal{D}$	Overall collection of documents (POI descriptors).
U	User profile
N_U	No. of POIs available, as preference history, in user profile U
D	Document (bag-of-words) representation of a POI, $D \in \mathcal{D}$
P	3-tuple representation of a POI, (D, T, r)
T	A set of user created tags, a subset of $\mathcal{T}$
r	User assigned rating for D, $r \in [0, 1]$
$\mathcal{T}$	Overall (controlled) vocabulary of tags used across the user profiles
l_U	*Hard* location constraint of U, $l_U \in L$
q_U	*Soft* contextual constraint(s) or trip qualifier(s) of U, $q_U \in Q$
$Q = Q_1 \times \ldots Q_c$	Overall set of non-location (*soft*) trip-qualifiers comprised of c trip qualifier types across the collection
Q_i	A particular non-location type constraint
$L(d)$	Location of a POI d
$M(\theta_U, q_U, l_U)$	Top set of M documents (location constrained to l_U) retrieved with the query with term distribution θ_U, q_U
$S(l_U)$	Set of POIs constrained to (*hard*) location, l_U
$\phi(P, d)$	Text-based content matching between a candidate POI d, and a POI $P = (D, T, r) \in U$
$\mathcal{S}(d, U)$	Text-based content matching between a candidate POI d, and the user profile U
$\psi_s(w, q_U)$	Contextual appropriateness measure of the term $w \mapsto [0, 1]$ for a *single context*, q_U
$\psi_j(w, q_U)$	Contextual appropriateness measure of the term $w \mapsto [0, 1]$ for a *joint context*, q_U

Table 3.1: List of the notations used in this thesis.

estimate the relevance scores,

$$\phi : U \times S(l_U) \mapsto \mathbb{R},\ S(l_U) = \cup\{d : L(d) = l_U\}, \tag{3.2}$$

where the output of the function, ϕ (e.g. with BM25 or a pseudo-relevance feedback method), does not depend on the non-location type qualifiers $q_U \in Q$.

A simple content matching technique is then to employ a standard ranking function, e.g. BM25, or language model computing the similarity between a candidate POI $d : L(d) = l_U$ and all POIs in the user profile,

$$\mathcal{S}(d, U) = \sum_{P=(D,T,r)\in U} \phi(P, d),\ d \in S(l_U), \tag{3.3}$$

where $\mathcal{S}(d, U)$ is the text-based content matching score between a candidate POI d, and the user profile U. Each POI in the current location context can then be sorted in descending order of their similarity

Categories	Values
Q_1: `trip-type`	{`business`, `holiday`, `other`}
Q_2: `trip-duration`	{`day-trip`, `longer`, `night-out`, `weekend-trip`}
Q_3: `accompanied-by`	{`alone`, `family`, `friends`, `other`}

Table 3.2: Soft constraint categories with their values.

scores and presented to the user.

Figure 3.1 shows a pictorial representation of our proposed IR setup for contextual suggestion where each POI is represented as a document (bag-of-words). A sample profile $P_i = (D, T, r)$ for a user's preference history is shown as a collection of three components (tuples): the document representation (D) of the POI, a set of tags (T) and the rating (r), provided by the user, for the POI. From the ranking perspective, we then need to perform content matching between a candidate document (representation of a candidate POI) $d : L(d) = l_U$ and every document (representation of profile $P_i = (D, T, r) \in U$) in the user's preference history.

We will see how we can model multiple *soft* contextual constraints, in addition with the *hard* location constraint, later in Chapter 6.

3.2 Experimental Setup

Our experiments are conducted with the TREC Contextual Suggestion[1] (TREC-CS) 2016 Phase-1 task [43]. The task requires a system to return a ranked list of 50 POIs (from a pre-defined collection) that best fit the user preference history and the user's current context. The (query) context is comprised of a *hard* location constraint, and $c = 3$ different non-location type *soft* qualifiers outlined in Table 3.2.

We now define the POI and user profile representation, followed by a detailed description of the data sets used for our experiments.

3.2.1 Representation of POIs and User Profiles

In our experimental setup, each document $D \in \mathcal{D}$ is represented as a bag-of-words which is comprised of descriptive information about the POI (available as a part of the crawled TREC web corpus) and other available information such as review texts collected from a location based social network (LBSN), viz. Foursquare. The combined use of the web crawl and content collected from LBSN as a static corpus complies with the standard experimental setup of most systems which participated in the TREC contextual suggestion (TREC-CS) tracks over a number of years [43].

[1] `https://sites.google.com/site/treccontext/`

We note at this point that since the crawled web content is likely to have been substantially different across different systems participating over a number of years in the TREC-CS tracks (primarily due to the dynamic nature of the content present in different LBSNs, and also because of changes in the APIs used to obtain the data). Consequently, the results reported by different TREC-CS participating systems are somewhat difficult to compare against one another. Instead of directly comparing against the reported results from the TREC-CS track overview papers, to ensure reproducibility and fairness in comparison of results, we apply a number of approaches within the same experimental framework.

Moreover, a majority of the TREC-CS participating systems made use of external data, such as ratings from other users, category information, external review texts etc. for their experimental setup. These systems, therefore, depend heavily on a number of different LBSN data sources, such as Trip Advisor, Yelp, Foursquare etc., which again makes the results difficult to compare due to the dynamic nature of the data and the APIs. To overcome reproduciblity and fairness concerns, our experiment setup makes use of a static data collection of POI content. Moreover, while it may be argued that applying a combination of post-processing techniques such as rule based heuristics developed from external knowledge resources [10], may further enhance the effectiveness of the methods investigated (including our proposed approaches), we do not employ any post processing techniques in our experiments. This is primarily because the purpose of our experiments is to investigate the effectiveness of different POI retrieval approaches under a data-driven controlled setup, and relying on a set of pre-existing rules defeats the purpose, because these rules are prone to changes with changes in the data, thus making such rule-based approaches not scalable.

For all our experiments, we only use a part of the user profile information, specifically, the POIs with a user-assigned rating higher than a threshold value. In the TREC-CS 2016 data, ratings are integers within $[-1, 4]$. As per the general user profile representation (Equation 3.1), each rating value is normalized within $[0, 1]$ (by min-max normalization). We then apply a threshold of 0.8 to define the *relevant* set of POIs for a user, i.e., these are the ones that are eventually used to construct the user profile for our proposed models. Formally speaking, in our experiments, the user profile U (Section 3.1.1) is comprised of only those triples, of the form (D, T, r), where $r \geq 0.8$. We selected this threshold value of 0.8 after a round of initial experiments, which is consistent with the instructions provided by TREC-CS 2016 task organizers.

3.2.2 Dataset

TREC-CS 2016 Data One of the reasons why we follow the TREC-CS 2016 framework [43] is that this framework, unlike others, facilitates a Cranfield-style evaluation with pool based relevance judgements, which is a key component in *Cranfield tradition* research paradigm [27], makes it a better choice for our experiments over other frameworks/datasets such as Yelp dataset[2].

A static web crawl of the TREC-CS 2016 collection has been released by TREC. There are around 1.2

[2] https://www.kaggle.com/yelp-dataset/yelp-dataset

Information	Value
Total number of POIs in corpus	1,235,844
Number of cities per user profile	1 or 2
Number of rated POIs per user profile	30 or 60
Total number of candidate cities	164
Number of candidate cities used by TREC	48
Maximum number of POIs per city	23,939
Minimum number of POIs per city	1,070
Average number of POIs per city	4,543.54
Total number of user profiles	438
Number of user profiles used by TREC	61

Table 3.3: TREC-CS 2016 [43] collection statistics.

million POIs in the TREC-CS 2016 collection that are based on 164 seed cities, out of which 48 of these seed cities were officially considered by TREC for experiments. Although the collection has a total of 438 user profiles, TREC officially used 61 profiles for the Phase-1 task, and released corresponding relevance assessments for these 61 user profiles. Table 3.3 shows a brief statistics of the TREC-CS 2016 collection. In each user profile, preference history is available for 1 or 2 seed cities with 30 or 60 POIs (i.e. 30 POIs per city), that have been rated by the user. Technically, a system needs to make contextual suggestion from a total of 48 seed cities for those 61 user profiles.

Fig. 3.2a shows a sample user profile with user ID.: 700, which has a set of POIs that the user visited in the past. The user rated the POI ‘TRECCS-00086310-160’ with rating 4, and assigned a tag “city walks”, for instance. This user profile also contains the user’s current contextual information such as the city identifier 359, which maps to city Billings, MT, USA, as the *hard* location context, and other non-location type *soft* contexts such as `trip-type=holiday`, `trip-duration=weekend-trip`, and `accompanied-by=family`. A sample document representation of a POI is shown in Fig. 3.2b, which follows the traditional TREC document format [42]. Each document is constituted of a ‘DOCNO’ field representative of its unique document (POI) identifier, and a ‘CITY’ field containing the city identifier of the POI. The city identifier 174 (of this example) as per the dataset [43] maps to Charlotte, NC, USA, which is the actual location of the POI. A ‘TEXT’ field representing the description of the POI contains the main content i.e. the descriptive texts about the POI, and/or the available review texts. Similarly, each rated POI available in the user profile such as the POI with ID ‘TRECCS-00086310-160’ in Fig. 3.2a has its unique document representation.

Fig. 3.2c, and 3.2d show a sample relevant (contextually appropriate), and a sample non-relevant document, respectively for the user profile (ID.: 700). The non-relevant POI (ID: TRECCS-00571606-359),

```
{ "id": 700,
    "group": "Family",
    "trip_type": "Holiday",
    "duration": "Weekend trip",
    "location": 359,
      "preferences": [
        { "rating": 4,
          "documentId": "TRECCS-00086310-160",
          "tags": ["city walks"]
        },
        { "rating": 3,
          "documentId": "TRECCS-00086622-160",
          "tags": ["cafés"]
        },
        { "rating": 2,
          "documentId": "TRECCS-00086333-160",
          "tags": ["deep sea fishing", "dolphin watching"]
        },
        .
        .
        .]
}
```

(a) User profile (ID: 700)

```
<DOC>
<DOCNO> TRECCS-00000106-174 </DOCNO>
<CITY> 174 </CITY>

<TEXT>
Common Market
The Common Market offers fresh deli sandwiches, cold beer, an
extensive wine selection, urban provisions, live music, knicks &
knacks and a neighbourhood connection.

CM South End is much the same as the original location; meaning
great draft and bottle beer, a tasty deli, and a fun casual
atmosphere. Great place to meet with friends. Have a pint or two,
have a glass of wine, take home a six pack or bottle. Awesome music
to rock to also! Great scene with all types of people and a cool
patio for nice weather :-) Dog friendly outside patio!

.
.
.

</TEXT>
</DOC>
```

(b) Document representation of POI (ID: TRECCS-00000106-174)

```
<DOC>
<DOCNO> TRECCS-00003664-359 </DOCNO>
<CITY> 359 </CITY>

<TEXT>
Cracker Barrel Old Country Store
Visit Cracker Barrel Restaurant and Old Country Store, where
pleasing people with our delicious homestyle cooking; gracious
service defines our country spirit.

Great service and excellent atmosphere. I would recommend going for
Friday Fish Fry. Friendly service. Trout amazing, wonderful service;
nice montana hospitality, if you dont enjoy it; your probably in the
wrong state The gaps in the mens room are at an acceptable level. No
sweet potato, miniature meatloaf and only biscuits for one of us.
The fireplace is awesome in the winter Try the Grandma's Sampler, it
isn't on the menu but it is the best mix of breakfasts foods. The
pork chops on Monday night are to die for!!!!! Great food at good
prices, but watch out for the fat content The gaps in the women's
bathroom stalls are too big. You can't get a better deal for
breakfast anywhere. Cracker Barrel is great! Buy the Billings
Gazette here

.
.
.

</TEXT>
</DOC>
```

(c) POI (ID: TRECCS-00003664-359) relevant to user (ID: 700)

```
<DOC>
<DOCNO> TRECCS-00571606-359 </DOCNO>
<CITY> 359 </CITY>

<TEXT>
The Vig Alehouse & Casino
The food is above average for a bar. Seems that the owners have a
constant turn over. American Restaurant Other Nightlife Sports Bar.
Your Heights Hide-a-way for great food, drinks and sports.

The Monte cristo is delicious. A nice balance of sweet and salty! I
also enjoy their garlic fries. Food is good, except for the
hamburgers. Trivia is also fun, I can see how others might not like
it since we usually win. They won the chowder competition!!! Awesome
food, great service, and specials! Bingo Wednesday nights! Love this
place Food, service, and atmosphere was great! I would highly
recommend the prime rib! Get the pork chop sandwich, better than Pug
Mahons! Meeting the cousins out west for the first time! Try your
luck in our casino and check out Vig rewards! Gamble on your
favorite NFL and College footballs games here, boards posted weekly!
Prime rib on the weekends is very good! Beer battered cod fish n
chips is awesome!!! Try the Black & Tan Onion Rings!! Best grilled
cheese ever!

.
.
.

</TEXT>
</DOC>
```

(d) POI (ID: TRECCS-00571606-359) not relevant to user (ID: 700)

Figure 3.2: Sample user profile (user ID: 700), and document representation of POIs from TREC-CS 2016 collection.

which is essentially a bar and/or casino, is possibly more appropriate when the user is with her friends. Note that both the relevant POI (ID: TRECCS-00003664-359), and the non-relevant POI (ID: TRECCS-00571606-359) are in the same city (ID: 359). However, the POI (ID: TRECCS-00000106-174) is in a different city (ID: 174), hence obviously non-relevant for the user of this example.

Details of the resource for modeling *soft* contextual constraints To incorporate non-location type qualifiers (*soft* constraints), one needs to learn an association between a word from the review text or the tag vocabulary of a user profile, and the likely (historical) context (trip-type, duration, etc.) that leads to creating the review text in the first place. A computational approach to automatically constructing this association requires the use of a knowledge base (e.g. a seed set of term-category associations).

#Assessors	Appropriateness	Term/Phrase	Single Context (Q_i)
12	1.00	American Restaurant	`trip-duration=weekend-trip`
7	0.71	American Restaurant	`trip-duration=longer`
12	-0.48	Nightlife Spot	`trip-type=business`
7	-1.0	Nightlife Spot	`accompanied-by=family`

Table 3.4: Crowd sourced contextual appropriateness data for *single context* [3]. Table 3.2 lists the categorical values corresponding to the three trip qualifiers.

#Assessors	Appropriateness	Term/Phrase	$Q = Q_1 \times Q_2 \times Q_3$ (`trip-type`, `trip-duration`, `accompanied-by`)
3	1.0	Movie Theater	‘`holiday, day-trip, friends`’
3	1.0	Irish Pub	‘`holiday, night-out, friends`’
3	1.0	Steakhouse	‘`business, longer, family`’
3	-1.0	Bar	‘`holiday, weekend-trip, family`’
3	1.0	Bar	‘`holiday, weekend-trip, alone`’
3	-1.0	Grocery Store	‘`business, day-trip, alone`’

Table 3.5: Crowd-sourced contextual appropriateness data for *joint context* [3].

Aliannejadi et al. [3] released a manually assessed dataset[3] comprising two different types of knowledge bases for information corresponding to a seed set of term-context associations.

Single context based appropriateness scores of some instances of association between a term or a short phrase and a single context are shown in Table 3.4. The appropriateness scores lie within $[-1, +1]$, -1 being completely inappropriate and $+1$ being completely appropriate. The first row of Table 3.4 shows that 12 assessors agreed that ‘American Restaurant’ is appropriate for the ‘trip-duration=Weekend-trip’ context. The average appropriateness score (from 7 assessors) for ‘American Restaurant’ is 0.7142 when the context is ‘trip-duration=Longer’. ‘Nightlife Spot’, as expected, is judged to be inappropriate for ‘accompanied-by=Family’.

For the joint context based appropriateness measure (Table 3.5), the scores are either -1 or $+1$, $+1$ being contextually appropriate and -1 being contextually inappropriate. It can be seen that an ‘Irish pub’ or a ‘Movie Theater’ is very appropriate (appropriateness score of 1.0), when a user is accompanied by her friends on a holiday trip. Similarly a ‘Steakhouse’ is appropriate when the joint context is business trip (trip-type), family (accompanied-by) and longer trip (trip-duration). Although a ‘bar’ is appropriate for the joint context “Holiday, Alone, Weekend trip”, it is judged to be inappropriate in the context of a

[3] Available at `https://www.inf.usi.ch/phd/aliannejadi/data.html`

weekend trip with family.

The contextual appropriateness data contains a total of 11 different contextual categories - 3 instances of ‘trip-type’ context (business trip, holiday or other trip), 4 instances of ‘trip-duration’ context (day trip, longer, night out or weekend trip), and 4 instances of ‘accompanied-by’ context (alone, family, friends or other). Assessments are available for 179 most frequent Foursquare category tags and 27 unique combinations of three contextual constraints. For our experimental setup, we normalized the contextual appropriateness scores for both single and joint context, within $[0, 1]$.

For the initial version of our proposed factored relevance model (Chapter 4), which addresses the location context only, we only make use of TREC-CS 2016 data. We will explain how we can transform the *soft* constraints into term weighting functions for POI retrieval by leveraging this small set of context-term annotations later in Chapter 6.

Chapter 4

Factored Relevance Model

The key idea of our proposed methodology for contextual recommendation (CR) is to make use of a pseudo-relevance feedback based framework to effectively balance the trade-off between exploitation and exploration. In this section, we first introduce the general concept of the relevance model. We then discuss how pseudo-relevance feedback in the form of a generalized relevance model can be applied in our problem context.

4.1 Relevance Model for IR

As discussed earlier in Chapter 2, a well known PRF method relevance model (RLM) [51] essentially estimates a term weight distribution $P(w|R) \approx P(w|Q)$, for a given query $Q = \{q_1, \ldots, q_n\}$. It is assumed that $P(w|R)$ also generate the set of terms in the top-M documents $\mathcal{M} = \{D_1, \ldots, D_M\}$, i.e.,

$$P(w|R) \approx P(w|Q) = \sum_{D \in \mathcal{M}} P(w|D) \prod_{q \in Q} P(q|D) \tag{4.1}$$

From Equation 4.1, it is evident that a high $P(w|Q)$ value (RLM term weight) results when a term w occurs frequently in a top-retrieved document (large $P(w|D)$ value) in conjunction with the frequent occurrence of a query term $q \in Q$ within D.

Each mention of 'relevance model' or 'RLM' in this thesis is to be interpreted as its more effective mixture model variant, i.e. 'RM3' [48].

4.2 User Profile based RLM

The primary challenge in matching a user profile with a POI descriptor in the current context (Equation 3.3) is to extract a set of contextually relevant terms from the documents and tags of the user profile. A naive way to compute the similarity scores in Equation 3.3 is to consider each document along with the user tags as a simple bag-of-words representation. This could potentially lead to noisy similarity estimation. To be more precise, there are two likely reasons that this naive similarity estimation may be ineffective. First, the information present in a user profile may be quite diverse in nature with only a specific aspect of it being likely to be useful in the current context, e.g. a user is likely to visit many different locations under different contexts in her past, however only a small number of them would be relevant within a present context. Second, it is often the case that the POI descriptors are long documents likely to introduce noise in the estimated similarities. Instead, focusing on relevant parts of these documents that are contextually related with the query rather than the whole document may lead to better similarity estimation.

With this motivation, we propose to employ a RLM to estimate a weighted distribution of terms extracted from the user profile, and use this term distribution θ_{U,q_U} to rank the POIs (documents) in the current context, $(l_U, q_U) \in (L, Q)$, where l_U is user's current location qualifier and q_U is the non-location type trip qualifier.

To estimate a relevance model based on a user profile U, we consider the set of tags in a POI descriptor $P = (D, T, r) \in U$ (Equations 3.3) as the observed or known terms (which are analogous to query terms in the IR framework of RLM). Let T' be the set of user assigned tags, i.e. union of all Ts from the set of tuples $(D, T, r) \in U$. A sample set $T' = \{$`American-restaurant, beer, beach, café, fast-food, shopping-for-wine`$\}$ is shown in Fig. 4.1. The set of top ranked documents on this occasion is the provided set of documents in the user preference history, i.e. union of all Ds from the set of tuples $(D, T, r) \in U$. Formally,

$$P(w|\theta_{U,q_U}) = \sum_{(D,T,r)\in U} rP(w|D) \prod_{t\in T'} P(t|D), \tag{4.2}$$

where the estimated RLM captures the semantic relationship between a user specified tag and a term presented in the documents, by co-occurrence corroboration from the user profile.

The rating values are used as confidence scores for the co-occurrences allowing the relevance model to assign higher weights to terms that co-occur more frequently with the user assigned tags within a POI with a high rating. Although it may seem at a cursory glance that the use of user assigned ratings in an RLM framework makes it supervised, we would like to emphasize that these rating scores are not used as *labels* in a supervised setting to optimize an objective function. In the degenerate case, i.e. when no ratings are available, our RLM-based feedback model would use a constant confidence value of 1, i.e. it would assign uniform weights to all POIs in the user profile.

4.3 Factored RLM for Contextual Relevance

To impose the *hard* constraint of the location qualifier l_U, we estimate another relevance model θ_{U,q_U,l_U}, by making use of both the user profile based relevance model estimated only with the soft constraints (Equation 4.2) and the selected subset of location-specific POIs (documents). This time the terms estimated in the user profile based RLM θ_{U,q_U} are considered to be the observed terms and the set of top ranked documents are the documents, denoted by $M(\theta_U, q_U, l_U)$, are top M documents retrieved in re-

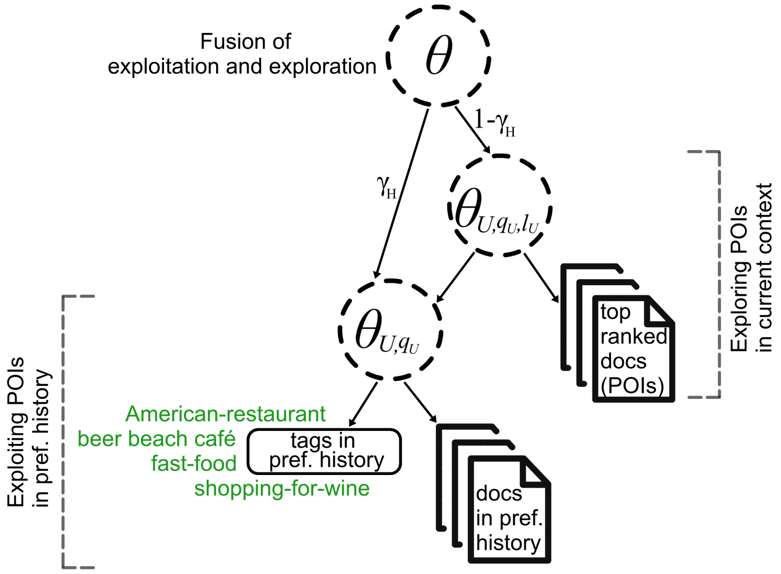


Figure 4.1: Schematic Diagram of a Factored Relevance Model (FRLM). It estimates a relevance model based on the user's preference history first (*exploitation*). Then it estimates another relevance model based on both the initial model and the top retrieved POIs in the current context (*exploration*). Finally these two models are linearly combined (*fusion*).

sponse to the query constrained to be satisfying the *hard* location constraint l_U. This is stated formally in Equation 4.3.

$$P(w|\theta_{U,q_U,l_U}) = \sum_{d \in M(\theta_U,q_U,l_U)} P(w|d) \prod_{t \in \theta_{U,q_U}} P(t|d). \tag{4.3}$$

Equation 4.3 is a factored relevance model in which estimating θ_{U,q_U,l_U} needs θ_{U,q_U} to be estimated first, which acts as the factor model. This factored relevance model *explores* the potentially relevant POIs in the user's current location context l_U, to achieve a better ranking of the POIs.

As a generalization, we use a linear combination of the two relevance models of Equations 4.2 (*exploitation* part) and Equation 4.3 (*exploration* part), into a combined model,

$$P(w|\theta) = \gamma_H P(w|\theta_{U,q_U}) + (1-\gamma_H) P(w|\theta_{U,q_U,l_U}), \tag{4.4}$$

where γ_H is the trade-off parameter to control the relative importance of the two relevance models. We call this version of our proposed model the Factored ReLevance Model (FRLM).

4.4 Algorithmic Details

As shown in Figure 4.1, our proposed methodology requires estimating a total of three relevance models. First, the user profile based relevance model, namely θ_{U,q_U}, is estimated by making use of the text present in a user's preference history. Next, to capture the relevance of POIs in a given context (typically a location of a POI), we estimate the factored RLM θ_{U,q_U,l_U} by using information from both the user preference and the top retrieved POIs, treating the former as equivalent to a query and the latter as top retrieved documents within an RLM framework [51]. Finally, both relevance models θ_{U,q_U} and θ_{U,q_U,l_U} are linearly combined into a single relevance model θ, the proposed generalized FRLM.

Algorithm 1: Proposed Algorithm using FRLM

```
Input: $U$, $\mu_U$, $\mu_{l_U}$, $M$ $\tau$, $\gamma_H$
Output: Ranked list for $U$
// Initialization
$T' \leftarrow$ Union of all $T$s $\in U$ // user assigned tags
$M(U) \leftarrow$ Union of all $D$s $\in U$ // POIs in pref. history
// Estimate relevance model $\theta_{U,q_U}$ from $M(U)$
$i = 0, L_U \leftarrow null, L'_U \leftarrow null$
for each term $w \in M(U)$ do
    $P(w|\theta_{U,q_U}) = 0$
    for each $P_i = (D,T,r) \in U$ do
        $P(w|\theta_{U,q_U})$ += $rP(w|D)\prod_{t\in T'} P(t|D)$
    end
    $P'(w|\theta_{U,q_U}) = \mu_U P(w|\theta_{U,q_U}) + (1-\mu_U)P(w|T')$
    $L_U[i++] \leftarrow \{w, P'(w|\theta_{U,q_U}\}$
end
Sort $L_U$ in descending order of $P'(w|\theta_{U,q_U})$
$L'_U \leftarrow$ top $\tau$ terms from $L_U$
$M(\theta_U, q_U, l_U) \leftarrow$ Top $M$ POIs from retrieve($L'_U$) // top-ranked POIs
// Estimate relevance model $\theta_{U,q_U,l_U}$ from $M(\theta_U, q_U, l_U)$
$i = 0, L_{l_U} \leftarrow null, L'_{l_U} \leftarrow null$
for each term $w \in M(\theta_U, q_U, l_U)$ do
    $P(w|\theta_{U,q_U,l_U}) = 0$
    for each document $d \in M(\theta_U, q_U, l_U)$ do
        $P(w|\theta_{U,q_U,l_U})$ += $P(w|d)\prod_{t\in\theta_{U,q_U}} P(t|d)$
    end
    $P'(w|\theta_{U,q_U,l_U}) = \mu_{l_U} P(w|\theta_{U,q_U,l_U}) + (1-\mu_{l_U})P(w|\theta_{U,q_U})$
    $L_{l_U}[i++] \leftarrow \{w, P'(w|\theta_{U,q_U,l_U})\}$
end
Sort $L_{l_U}$ in descending order of $P'(w|\theta_{U,q_U,l_U})$
$L'_{l_U} \leftarrow$ top $\tau$ terms from $L_{l_U}$
// Estimate generalized factored relevance model $\theta$
$i = 0, L \leftarrow null$
for each $w \in$ Union of $(L'_U, L'_{l_U})$ do
    $P(w|\theta) = \gamma_H \cdot P'(w|\theta_{U,q_U}) + (1-\gamma_H) \cdot P'(w|\theta_{U,q_U,l_U})$
    $L[i++] \leftarrow \{w, P(w|\theta)\}$
end
// Query expansion
$T'' \leftarrow$ all terms in $L$ // weighted term distribution
retrieve($T''$) // ranklist for U
```

4.4.1 Proposed Algorithm

Initialization As described in Algorithm 1, let each user profile $U = \cup_{i=1}^{N_U}\{P_i : P_i = (D,T,r)\}$ as formalized in Equation 3.1. For each user U, we construct a set, T', which is comprised of all tags used by the user. In other words, a union of all Ts from the set of tuples $(D, T, r) \in U$ is stored in a set T'. Then, we collect all document representations, i.e. union of all Ds from the set of tuples $(D, T, r) \in U$ and store them in set $M(U)$.

Estimation of User Profile based RLM Now we estimate the user profile based relevance model θ_{U,q_U}. For each term $w \in M(U)$, we compute the probability of sampling the term w from θ_{U,q_U}, denoted by $P(w|\theta_{U,q_U})$, as in Equation 4.2 by the joint probability of observing w along with the tags T'. Please note that we take a mixture model of the estimated relevance model θ_{U,q_U} in conjunction with the tags likelihood model i.e. $P(w|T')$, to get the influence of tags T' in the final estimation of θ_{U,q_U}. Tags likelihood model $P(w|T')$ is computed using maximum likelihood estimation (MLE). Smoothing parameter μ_U (tag mixing) controls the relative weight we assign to the relevance model versus the 'tags' model.

$P'(w|\theta_{U,q_U})$ is the final probability of term w from the mixture model. The term w is added in an initially empty list L_U along with its probability $P'(w|\theta_{U,q_U})$. When every term in $M(U)$ is considered, the list L_U is sorted in descending order of $P'(w|\theta_{U,q_U})$ and top τ terms are added in another initially empty list L'_U, along with their corresponding $P'(w|\theta_{U,q_U})$ values. We then execute an initial retrieval with the terms in set L'_U. We store top set of M documents (location constrained to l_U) retrieved with the query with term distribution θ_U, q_U, i.e. L'_U.

Estimation of Factored RLM We then estimate the factored RLM θ_{U,q_U,l_U} from the set $M(\theta_U, q_U, l_U)$, to capture the contextual relevance in a similar fashion. As described in Algorithm 1, here $P(w|\theta_{U,q_U,l_U})$, i.e. the probability of sampling a term w from θ_{U,q_U,l_U}, is computed by the joint probability of observing w along with the terms in the previously estimated user profile based RLM θ_{U,q_U}. We get a list L'_{l_U} in the same way we created the list L'_U.

Generalization of FRLM We then linearly combine two lists L'_U and L'_{l_U} into the final list L with a smoothing parameter γ_H. Finally all terms in L, along with their corresponding probabilities are put in set T'', an expanded set of tags (terms) which is analogous to expanded query in IR. Note that T'' can be considered as a weighted query as it is a distribution of terms along with their probabilities, i.e. each term in the distribution is boosted by its probability as the weight of the term. Finally, we execute another retrieval with T'' to get the final result and present it to the user U.

4.5 Methods Investigated

Following the experimental setup described in Section 3.2 (Chapter 3), we employ a number of standard IR based and recommender system (RecSys) based methodologies as baselines for comparison against our proposed models. In addition to investigating the overall effectiveness of alternative approaches, with respect to our proposed models, we particularly focus on finding an optimal trade-off between a user's preference history (*exploitation*) and the information about the POIs constrained to a *hard* contextual constraint such as 'location' (*exploration*) for contextual POI recommendation.

4.5.1 IR Baselines

To acquire the comparative effectiveness of our proposed approaches we choose a number of baselines based on ablations of components/factors from our proposed models. The IR baselines are enlisted below.

1. **BL1 - BM25**: We employ the standard BM25 retrieval model as the similarity measure function of Equation 3.3. We select user assigned tags (T', as we used in Equation 4.2) from the set of tuples (D, T, r), where $r \geq 0.8$ to form the query. BM25 parameters k, b are optimized by grid search with respect to nDCG@5.

2. **BL2 - Term Selection**: Since our proposed models estimate a weighted term distribution, we apply a method of extracting a set of terms from the set of documents from the set of tuples (D, T, r), where $r \geq 0.8$ (based on BM25 weights) as one of the baselines. Note that the parameter settings of k and b for BM25 remain the same as that of BL1. We optimize the number of selected terms to 25 by grid search. This model is able to take into account exploitation by selecting terms from user profile.

3. **BL3 - BM25 with Term Selection:** Since we combine both the user preference history and information about the POIs within a current context for FRLM estimation, we apply a CombSUM [88] technique to merge the two ranked lists obtained with BL1 (BM25) and BL2 (Term Selection). This offers a naive method of combining two sources of information, i.e. user preference history and the POI content in current contexts.

4. **BL4 - RLM**: Since, at its core, our proposed approach relies on estimating a factored relevance model, we select the traditional relevance model (RLM) of Equation 4.1 as a baseline. Similar to BL1 (BM25), we consider the user assigned tags from the user profile with ratings $r \geq 0.8$ as observed terms (analogous to a query). We then estimate a relevance model (RLM) to rank the POIs within the current context. In contrast to the factored relevance model, this baseline model only makes use of the *exploration* part while formulating the query, i.e., with respect to the standard RLM [51], the set of tags in a user history acts as the query and the RLM term weights are computed using the local co-occurrences from the top-retrieved POI descriptors constrained to a given user-specified location.

5. **BL5 - KDERLM**: We choose word vector compositionality based relevance feedback using kernel density estimation [84] as another baseline. This baseline corresponds to a KDE based generalized version of traditional RLM (the factored part corresponding to an enriched matching between the user profile and POIs in a current location being ablated). Similar to BL4 (RLM), in this baseline we also use the tags from a user profile with ratings higher than or equal to 0.8 as observed terms (analogous to a query), and then estimate a KDE-based RLM to score POIs within a current context.

Parameters for each method were separately tuned with the help of a grid search. Since our proposed models are unsupervised (without involving any parameter learning with the help of gradient descent updates), we do not employ a separate train and test split for conducting grid search. Two common parameters to all the relevance feedback models are the number of feedback documents, M, and the number of feedback terms, τ. It was found after a grid search that RLM and FRLM yielded optimal results with the values 5 (#documents) and 25 (#terms). Similarly for KDERLM, M and τ were optimized to the values 3 and 80.

4.5.2 Recommender System Baselines

In the absence of other users' ratings, it is not possible to apply standard recommender system (RecSys) approaches such as, collaborative filtering directly to predict the relevance of a POI (considered as an item in RecSys research). However, a disparate analogy allows us to employ standard RecSys methodologies as a pre-processing step in our experimental setup. Specifically, one may imagine that the contents in user profiles are analogous to users in RecSys terminology, whereas the set of user-assigned tags used to describe POIs are analogous to items. This user-item analogy allows us to learn semantic associations between a user profile and the tag vocabulary. Given a user profile, it is thus possible to enrich the set of tags (analogous to suggesting more items for a user in the traditional framework of RecSys research). Following this general set up for our RecSys based experiments, we now explain the details of each RecSys based baseline approach.

6. **BL6 - Most Popular K**: A simple (but effective) RecSys methodology is the recommendation of the *most popular* items based on overall ratings across all users, with the expectation that these items will be appropriate to the new user as well [89]. With respect to our experimental setup, we extract the K most popular tags across each user's preference history. We then use these selected tags to form the query for each user. For instance, if the tag 'beer' is one of the most popular tags in the tag vocabulary across all users, suggesting pubs as candidate POIs for a new user is likely to be a good recommendation.

   After formulating an enriched query based on the most popular tags, we apply the standard BM25 retrieval model as the similarity matching function (Equation 3.3) with the same settings of k, b, as that in BL1. K (the number of popular tags to extract for enriching the query) is optimized based on the average rating of tags across the set of all users. The threshold for this average rating was set to 0.8.

7. **BL7 - Profile Popular K**: In contrast to the previous approach of finding the globally most popular tags across all users, this approach restricts the selection of the most popular tags to each user profile only. It can be argued that this approach extracts tags in an entirely personalized manner. For instance, this method selects the tag 'seafood' as a query term if it is one of the most popular tags in the preference history of only the current user. Similar to BL6 (Most Popular K), BM25 is used as the similarity function (Equation 3.3) with the same settings of k, b, as that in BL6. K is optimized based on the user profile specific average rating of tags and the cut-off for average rating is set to 0.8.

8. **BL8 - NeuMF**: We used a state-of-the-art neural network based matrix factorization method [45], which makes use of a fusion of generalized matrix factorization (GMF) and multi-layer perceptron

(MLP) to better model the complex user versus item interactions (in our case, an item corresponding to a tag). Similar to the Popular-K baselines (both collaborative and personalized), the K most likely tags, as predicted by the NeuMF model, are then used to construct a query.

9. **BL9 - Bayesian content-based recommendation**: A standard text classification based content matching technique, widely used in recommender systems, is employing a Bayesian classifier [69]. As per the requirement of a supervised binary classification approach, we consider the set of all positively rated documents in a user profile, i.e. all Ds from the set of tuples (D, T, r), where $r \geq 0.8$, as the 'positive' class, whereas the set of all negatively rated documents in a user profile, i.e. all Ds from the set of tuples (D, T, r), where $r < 0.8$, are considered to define the 'negative' class. We then train a binary Naive-Bayes classifier. During recommendation, for each POI that is classified as 'positive', we consider the posterior likelihood value of the classifier as the score of the POI. We then present the ranked list by sorting the POIs in decreasing order of these likelihood scores.

4.5.3 Hybrid Baselines

10. **BL10 - Content + Tag Matching**: As mentioned earlier in Section 3.2.1, due to the use of external data resources by the TREC-CS participating systems, the results reported therein are not directly comparable with our results (in terms of the absolute values of the measured metrics). We therefore conduct experiments with the recorded best performing method of TREC-CS 2016 within our setup. This method involves a hybrid of content and tag matching [2, 1]. More precisely speaking, the similarity matching function of this method is a combination of query words/tags and document (POI) words/tags similarity (Content + Tag) score.

11. **BL11 - Hybrid**: We employ a CombSUM [88] of the two ranked lists obtained with the best performing IR-based baseline BL5 (KDERLM), and another strong baseline BL10 (Content + Tag Matching), which allows provision for an ensemble of content and tag matching.

Distances between word vectors are used in the kernel density based approaches (i.e. the baseline KDERLM, and the KDE based generalization of FRLM which is described later in Chapter 5), and in modeling the soft constraints (i.e. multi-contextual generalization of FRLM which is described later in Chapter 6). Specifically, for our experiments the embedded space of word vectors is obtained by executing skipgram [68] with default values for the parameters of window-size (5) and the number of negative samples (5), as set in the `word2vec` tool[1]. Skipgram was trained on the collection of the POI descriptors in the TREC-CS collection.

[1] https://github.com/tmikolov/word2vec

		Graded Evaluation Metrics			Binary Evaluation Metrics			
	Method	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
IR-based approaches								
BL1	BM25	0.2747	0.2484	0.2889	0.3934	0.3066	0.1326	0.6539
BL2	Term Sel.	0.2484	0.2383	0.3034	0.3639	0.3066	0.1466	0.6148
BL3	BM25 + Term Sel.	0.2411	0.2332	0.3143	0.3672	0.3115	0.1530	0.5607
BL4	RLM [51]	0.2615	0.2453	0.3091	0.3574	0.3033	0.1437	0.6441
BL5	KDERLM [84]	0.2829	0.2682	0.3191	**0.3967**	0.3361	0.1495	0.6539
RecSys based approaches								
BL6	Most Popular K	0.1861	0.1926	0.2580	0.2787	0.2705	0.1016	0.4154
BL7	Profile Popular K	0.2488	0.2409	0.2811	0.3410	0.3016	0.1280	0.6486
BL8	NeuMF [45]	0.1626	0.1655	0.2480	0.2361	0.2344	0.0937	0.4314
BL9	Bayesian	0.2170	0.1774	0.1816	0.3082	0.2082	0.0672	0.5831
Hybrid approaches								
BL10	Content + Tag. [2]	0.2499	0.2411	0.2800	**0.3967**	0.3377	0.1330	0.5390
BL11	Hybrid (BL5 + BL10)	0.2805	0.2667	0.3329	0.3902	0.3311	0.1583	0.6514
Proposed approach								
	FRLM ($\gamma_H = 0.8$)	**0.2919**	**0.2810**‡	**0.3418***†‡	0.3934	**0.3443**‡	**0.1616***‡	**0.6786**

Table 4.1: Comparisons between POI retrieval approaches with location only (*hard*) constraint. The notations, ‘*’, ‘†’ and ‘‡’ denote significant (paired t-test with 95% confidence) improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid), and BL1 (BM25), respectively.

4.6 Results

Table 4.1 shows the results obtained by each contextual recommendation approach that we investigated, as outlined in Section 4.5. Each method was separately optimized with grid search on the nDCG@5 metric, the official metric to rank systems in the TREC-CS task. The table shows that FRLM outperforms all other baselines with respect to most standard evaluation metrics, except P@5. Improvements in nDCG are statistically significant at 95% confidence level based on the Wilcoxon signed-rank test. Significant improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25), respectively are shown in Table 4.1.

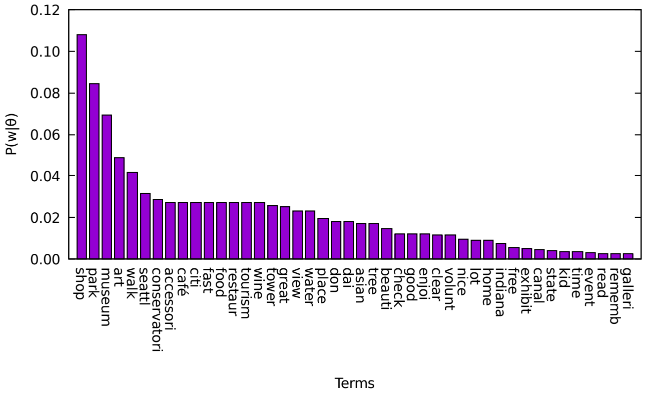


Figure 4.2: An instance of FRLM term distribution weights (sorted from highest to lowest) for location only modeling using $M = 5$ (number of top-retrieved documents for feedback as per the $M(\theta_U, q_U, l_U)$ notation of Table 3.1) and $\tau = 25$ (number of top-scoring terms in the estimated RLM distributions).

4.6.1 Factored Model (exploration and exploitation) Outperforms the Other Approaches

The superior performance of the factored model (FRLM) in comparison with BL2 (Term Selection) indicates that the probability distribution of weighted terms, as estimated by the factored models, is a more effective way to select candidate terms for query formulation. Although BL3 (BM25 + Term selection) takes both the preference history of the user (term selection based exploitation) and the top ranked POIs (BM25 based exploration) into account, the superior performance of FRLM indicates that such information turns out to be more effective when intricately integrated within the framework of a relevance based model, leveraging information from both preference history and the top retrieved POIs (rather than the ad-hoc way of first retrieval and then term selection for query expansion). Figure 4.2 shows FRLM term distribution for a user request (user ID 763) where $T' = \{$`art, cafés, city-walks, fast-food, museums, parks, restaurants, shopping-for-accessories, shopping-for-wine, tourism`$\}$. FRLM assigns higher weights to terms such as 'park', 'museum', which are clearly relevant for this particular example. Indeed, this model is also successful at capturing other relevant terms such as 'view', 'tree', 'canal' etc.

4.6.2 IR Approaches Outperform Collaborative/personal RecSys Ones

A common and sometimes very useful recommendation approach is BL6 (Most Popular K). The poor performance of this method demonstrates that globally popular items (across a number of different users) do not work well for the POI retrieval task. The likely reason for this being that personal choices in this case are more important. The fact that BL7 (Profile Popular K) performs better than BL6 is consistent with this hypothesis of emphasizing personal preferences more than the global ones.

However, it can be seen that the effectiveness of this RecSys based approach (BL7) is inferior to that of BL1 (BM25), which is a standard IR based approach making use of the information in the set of tags from

POIs with ratings higher than 0.8. This shows that user ratings are more important than the popularity (relevance likelihood) of tags created by a user. A frequently used tag may have been used to create negative reviews by a user, in which case assigning importance to these tags may introduce noise into POI recommendation.

4.6.3 Unsupervised Approaches Outperform Supervised Ones

Supervised approaches, namely BL8 (NeuMF) and BL9 (Bayesian), do not perform well. This is most likely due to the lack of sufficient training data. One of the problems of a supervised approach is that it involves learning a hard decision during the training phase to classify POIs as either relevant (with rating values higher than a threshold) or non-relevant (otherwise). The advantage of our proposed models is that they do not involve hard decision steps during any stage of their working procedure. Moreover, the primary advantage of an unsupervised approach is that it can work in situations where user preference data (for training) is sparse or even non-existent.

Another observation from the comparisons between FRLM and the matrix factorization based technique BL8 (NeuMF) is that representation learning over words (which is trained on unannotated document collections available in large quantities) is more beneficial than the matrix factorization based joint representation learning of users and POIs in a latent space (which requires large quantities of training data in the form of user-item associations). Moreover, the POI recommendation problem is more of a personalized retrieval problem, where information from other users (which is what happens in a user-item matrix factorization based collaborative setup such as NeuMF) may in fact turn out to be ineffective. This is also reinforced by our previously reported observation that ‘Profile Popular K’ (personalized retrieval) outperformed the ‘Most Popular K’ (collaborative retrieval).

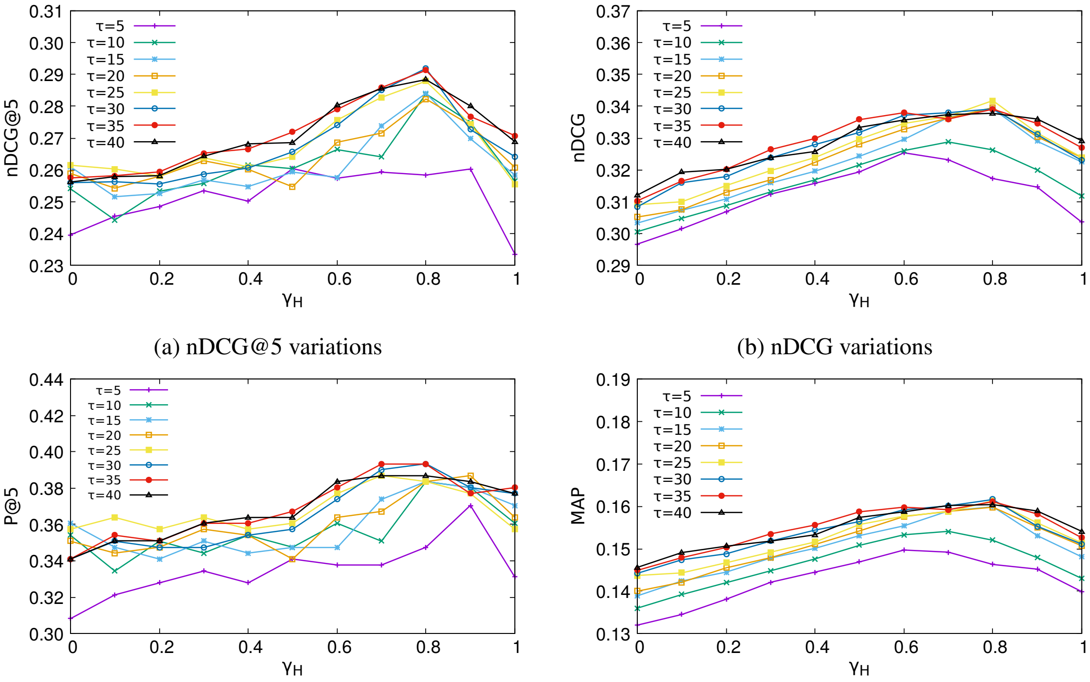
4.6.4 A Combination of Content and Tags is more Effective than Tag-matching Alone

The BL10 (Content + Tag matching) baseline involves a hard classification step, and then an aggregation over tag matching scores. In contrast, our proposed model (FRLM) does not involve hard selections for either documents or tags/terms, which means that they are able to selectively leverage the information from each source.

4.6.5 FRLM Sensitivity

Next, we investigate the effects of varying the parameter γ_H (i.e. the trade-off between exploration and exploitation) on the performance of FRLM. Figure 4.3 shows the sensitivity of FRLM (measured with nDCG@5, nDCG, P@5 and MAP) with respect to the number of terms (τ) used to define the weighted distribution of terms, and the relative importance of the historical context of a user with respect to the POIs in the current context, i.e. γ_H.

An interesting observation is that FRLM performs best with a balanced trade-off between exploration and exploitation. In particular, the optimal results (both in terms of nDCG@5 and nDCG) are obtained



(a) nDCG@5 variations

(b) nDCG variations

(c) P@5 variations

(d) MAP variations

Figure 4.3: Effect of precision at top ranks (nDCG@5 and P@5) and recall (nDCG and MAP) with respect to changes in number of terms used in FRLM estimation (τ) and the relative importance assigned to user profile information (γ_H).

when $\gamma_H = 0.8$. Moreover, the effectiveness of FRLM is not good with the user profile history only, which indicates the diverse nature of the historical information itself and demonstrates the usefulness of selectively extracting pieces of information from the history that are contextually relevant in the current situation.

We also observe that too few or too large a number of terms tends to decrease retrieval effectiveness. The former is not able to adequately capture the relevant semantics required to match the profile with the current context, while the latter introduces noise (from parts of profile that are not relevant in the current context) in the estimated FRLM distribution.

4.7 Summary

The challenge of providing personalized and contextually appropriate recommendations to a user is faced in a range of use-cases, e.g., recommendations for movies, places to visit, articles to read etc. In this chapter, we focus on one such application, namely that of suggesting ‘points of interest’ (POIs) to a user given her current location (*hard* context), by leveraging relevant information from her past preferences.

An automated contextual recommendation algorithm is likely to work well if it can extract information from the preference history of a user (*exploitation*) and effectively combine it with information from the user's current context (*exploration*) to predict an item's 'usefulness' in the new context. To balance this trade-off between *exploitation* and *exploration*, we propose a generic unsupervised framework involving a factored relevance model (FRLM), comprising two distinct components, one corresponding to the historical information from past contexts, and the other pertaining to the information from the local context.

Our experiments are conducted on the TREC contextual suggestion (TREC-CS) 2016 dataset. A characteristic of our model is that it achieves a sweet-spot between the user's preference history in past contexts (exploitation), and the relevance of top-retrieved POIs in the user's current context (exploration). Our experiments on the TREC-CS 2016 dataset show that our proposed model of a factored relevance model is able to effectively combine these two sources of information, leading to significant improvements in contextual recommendation quality.

Chapter 5

Word Semantics for POI recommendation

The user profile based RLMs as presented in Chapter 4 (θ_{U,q_U} of Equation 4.2 or its factored version, θ_{U,q_U,l_U}, of Equation 4.3) can take into account only the document level co-occurrence of terms (ignoring any semantic associations between them). In this chapter, we generalize the proposed factored relevance model of Chapter 4 by employing the concept of kernel density estimation. The primary motivation behind doing this is to achieve a better semantic match between the POI descriptions and the review/description text of the locations visited in the past by a user. In the context of our specific problem, this favours those terms which in addition to exhibiting local (top-retrieved) co-occurrence, are also semantically related to the query terms. Before describing our generalized model, we outline the existing work on kernel density based relevance models [84].

5.1 Kernel Density Estimation based RLM

Kernel Density Estimation (KDE) is a non-parametric method to estimate the probability density function of a random variable. Formally, let $\{x_1, \ldots, x_n\}$ be independent and identically distributed (i.i.d.) samples drawn from a distribution. The shape of the density function, f, from which these points are sampled can be estimated as

$$\hat{f}_\alpha(x) = \frac{1}{nh} \sum_{i=1}^{n} \alpha_i K\Big(\frac{x - x_i}{h}\Big), \tag{5.1}$$

where x_i is a given data point (commonly known as the pivot point), $\hat{f}_\alpha(x)$ is the estimated value of the true density function $f(x)$, α_i is the relative importance of the i^{th} data point with the constraint that $\sum_i \alpha_i = 1$, and $K(.)$ is a kernel function scaled by a bandwidth parameter h. By definition, a kernel function is a monotonically increasing function of the distance between two points (vectors). A common choice of a kernel function is a Gaussian function.

Roy et al. [84] observed that since the relevance model estimates a distribution of (real-valued) weights over terms, the concept of KDE can be applied to define this distribution in a generalized way (the model being called Kernel Density Estimation based RLM or KDERLM for short). The basic idea to define the relevance model distribution this way is to treat the query terms as a set of pivot terms (analogous to the x_i's of Equation 5.1). Rather than treating terms as independent, the distance between the vector representation (obtained by applying a word embedding method such as `word2vec` [68]) of a pivot (query) term with that of a term occurring in the top-ranked documents is then used to define the kernel function. This results in the influence of a query term *propagating* to other terms that have similar (close) vector representations in the embedded space. Formally, assuming that the query terms $Q = \{q_1, \ldots, q_n\}$ are embedded as vectors, the probability density function estimated with KDE is

$$f(w) = \frac{1}{nh} \sum_{i=1}^{n} P(w|\mathcal{M}) P(q_i|\mathcal{M}) K\Big(\frac{w - q_i}{h}\Big), \tag{5.2}$$

where the kernel function K is a function of the distance between the word vectors of a term w (within a top-ranked document) and a query term q_i. The set of top-ranked documents is considered as a single document model $\mathcal{M}$. Moreover, $P(w|\mathcal{M})P(q_i|\mathcal{M})$ acts as the weight associated with this kernel function (thus incorporating the local RLM effect in addition to the global term semantics from the embedded space). In other words, the closer the word w is to a query term q_i in conjunction with a high RLM term weight, the higher becomes the value of the KDERLM weight $f(w)$.

5.2 KDE based RLM on User Profiles

In the context of the POI recommendation problem, the KDERLM model potentially assigns higher importance to a word w from a POI descriptor if it is semantically associated to a tag (query) term t (as per the embedding space). We can imagine that the discrete probabilities $P(w|\theta_{U,q_U})$ of the user profile based RLM (Equation 4.2) are smoothed out to form a continuous probability density function $f(w)$. As seen in Figure 5.1, the shape of this density function is controlled by a set of pivot points comprising the tag terms in a user's profile. Concretely, for a user profile $U = \cup_{i=1}^{N_U}\{P_i : P_i = (D, T, r)\}$ with the set of unique tag terms, T', the probability density function estimated by KDE (with a Gaussian kernel) is given by

$$f_\alpha(w) = \frac{1}{nh} \sum_{t \in T'} \alpha_t K\Big(\frac{w - t}{h}\Big) = \sum_{t \in T'} \alpha_t \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\mathbf{w} - \mathbf{t})^T(\mathbf{w} - \mathbf{t})}{2\sigma^2 h^2}), \tag{5.3}$$

where $\mathbf{w}$ and $\mathbf{t}$ denote the vectors for the word w and the tag t, and α_t is the weight assigned to the tag term which we describe how to compute next.

Considering the set of all documents (reviews or POI descriptions) of a user profile, i.e. D belonging to some tuple in $U = \cup_{i=1}^{N_U}\{P_i : P_i = (D, T, r)\}$, as a single document model $\mathcal{M}$, the estimation of our previously proposed user profile based RLM (Equation 4.2) can be reduced as shown in Equation 5.4.

$$P(w|\theta_{U,q_U}) = P(w|\mathcal{M}) \prod_{t \in T'} P(t|\mathcal{M}) \tag{5.4}$$

Then maximum likelihood estimates (MLE) of $P(w|\mathcal{M})$ and $P(t|\mathcal{M})$ ensure that to maximize $P(w|\theta_{U,q_U})$, both $P(w|\mathcal{M})$ (i.e. the normalized term frequency of a word w in the set of documents in the user's preference history, or in other words, the set of terms a user generally prefers, e.g., 'friends', 'pubs' etc.), and

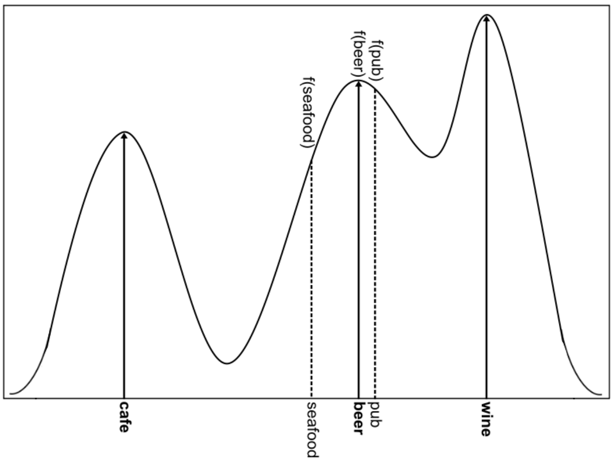


Figure 5.1: FRLM density estimation with KDE

$P(t|\mathcal{M})$ (i.e., the normalized term frequency of the tags in the set of documents in the user's preference history) are both maximized, i.e., Equation 5.4 captures the local co-occurrences between a tag and a term within a user profile. We then assign $\alpha_t = P(w|\mathcal{M})P(t|\mathcal{M})$ and substituting it in Equation 5.3, yields Equation 5.5.

$$f(w) = \sum_{t \in T'} P(w|\mathcal{M})P(t|\mathcal{M}) \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2 h^2}) \tag{5.5}$$

Figure 5.1 shows a schematic example of the KDERLM distribution estimated with three sample tag terms, such that the user profile based RLM probability distribution function can be visualized as a function pivoted around these three tag vectors projected on a line. In Figure 5.1, there are two terms 'seafood' and 'pub' in the neighbourhood of a tag term 'beer'. As 'pub' is closer to 'beer' than 'seafood', the value of the density function at 'pub', i.e. $f(pub)$, is higher than that at 'seafood', i.e. $f(seafood)$.

In Equation 5.5, we consider all documents in the user's preference history as a single document model and ignored document level user rating. To incorporate the document level importance of a term w in the estimation of the probability density function, we introduce the document level user rating while computing $P(w|\mathcal{M})$. We compute document-level user rating based relevance weights, $P(w|\mathcal{M})$ as shown in Equation 5.6.

$$P(w|\mathcal{M}) = \sum_{(D,T,r) \in U} rP(w|D) \tag{5.6}$$

Plugging this into Equation 5.5 yields Equation 5.7.

$$P(w|\theta_{U,q_U}; h, \sigma) = \sum_{t \in T'} \Big(\sum_{(D,T,r) \in U} rP(w|D) \Big) P(t|\mathcal{M}) \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2 h^2}) \tag{5.7}$$

Similar to our previous version of the user profile base RLM (Equation 4.2), the rating values in Equation 5.7 are used as confidence scores for the co-occurrences, which allows the relevance model to preferentially weigh the term co-occurrences of across POIs that are high rated in a user profile.

5.3 A Factored version of KDERLM

We argued in Section 4.3 (Figure 4.1) that a factored version of the RLM is particularly suitable for the task of contextual POI recommendation because it is useful to enrich the initial query (comprised of tag terms) with additional relevant terms from the user profile (review text/POI descriptors). Since term weights estimated from Equation 5.7 yield a set of such potentially relevant terms, we make use of the term weight distribution estimated from Equation 5.7 to estimate another relevance model for the retrieval step with the hard location constraint, i.e., this time the term weights are useful to effectively match the information need (weighted query estimated from a user profile) with the documents that are to be retrieved (specified by the set of documents from the collection satisfying the location constraint). More formally,

$$P(w|\theta_{U,q_U,l_U};h,\sigma) = \sum_{d \in M(\theta_U,q_U,l_U)} \frac{1}{\sigma\sqrt{2\pi}} P(w|d) \prod_{t \in \theta_{U,q_U}} P(t|d) \exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2 h^2}), \quad (5.8)$$

where we make use of the set of POIs of the current location (constrained by $L(d) = l_U$) to estimate the KDERLM corresponding to the *exploration* mode (similar to Equation 5.5).

Similar to FRLM, where we combine both the models corresponding to exploitation and exploration, we can create a combined version of this KDE based model as shown in Equation 5.9.

$$P(w|\theta;h,\sigma) = \gamma_H P(w|\theta_{U,q_U};h,\sigma) + (1-\gamma_H)P(w|\theta_{U,q_U,l_U};h,\sigma) \quad (5.9)$$

The trade-off parameter γ_H controls the relative importance of the two relevance models. We call this version of our proposed model Kernel Density Estimation based Factored ReLevance Model (KDEFRLM), scaled with kernel bandwidth h, and standard deviation σ.

5.4 Methods Investigated

Following the experimental setup described in Section 3.2 (Chapter 3), we employ the same set of IR-based, recommender system (RecSys) based, and hybrid methodologies as baselines, as we did for FRLM in Chapter 4, for comparison against our word embedding based model. In particular, **BM25**, **Term Selection**, (**BM25 + Term Selection**), **RLM** [51], and **KDERLM** [84] have been employed as IR-based baselines. On the other hand, we employ **Most Popular K**, **Profile Popular K**, **NeuMF** [85], and **Bayesian** content-based recommendation [69] as our RecSys baselines. Hybrid approaches include **Content + Tag** [2, 1], and a **hybrid** of KDERLM, and Content + Tag.

In addition to investigating the overall effectiveness of alternative approaches, with respect to our proposed model, we specifically focus on finding an optimal trade-off between a user's preference history (*exploitation*) and the information about the POIs constrained to a *hard* contextual constraint such as 'location' (*exploration*) for contextual POI recommendation.

Parameters for each method were separately tuned with the help of a grid search. As mentioned earlier for FRLM in Chapter 4, since our proposed models are unsupervised (without involving any parameter learning with the help of gradient descent updates), we do not employ a separate train and test split for conducting grid search. Two common parameters to all the relevance feedback models are the number of feedback documents, M, and the number of feedback terms, τ. It was found after a grid search that RLM and FRLM yielded optimal results with the values 5 (#documents) and 25 (#terms). Similarly for KDERLM, M and τ were optimized to the values 3 and 80, whereas for KDEFRLM, the optimal values of M and τ were found to be 2 and 100, respectively.

Distances between word vectors are used in the kernel density based approaches i.e. the baseline KDERLM, and the KDE based generalization of FRLM. Specifically, for our experiments the embedded space of word vectors is obtained by executing skipgram [68] with default values for the parameters of window-size (5) and the number of negative samples (5), as set in the `word2vec` tool[1]. Skipgram was trained on the collection of the POI descriptors in the TREC-CS collection.

5.5 Results

Table 5.1 shows the results obtained by each contextual recommendation approach that we investigated, as outlined in Section 5.4, which was in fact same for FRLM in Section 4.5 (Chapter 4). Each method was separately optimized with grid search on the nDCG@5 metric, the official metric to rank systems in the TREC-CS task. The table shows that KDEFRLM outperforms all other baselines with respect to most standard evaluation metrics. Improvements in nDCG, and MAP are statistically significant at 95% confidence level based on the Wilcoxon signed-rank test. Significant improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25), respectively are shown in Table 5.1.

5.5.1 Common Observations

Some observations that we already mentioned in Chapter 4, while experimented with FRLM such as, the factored models (exploration and exploitation) outperform the other approaches, IR approaches outperform collaborative/personal RecSys ones, unsupervised approaches outperform supervised ones etc. are common here. Following the similar trend in performance, KDEFRLM in fact outperforms the initial FRLM. In addition to improvement in recall (i.e. with respect to nDCG, and MAP), KDEFRLM achieves a better precision (nDCG@5, nDCG@10, P@5, and P@10). Figure 5.2 shows the comparison of relative term distributions (common terms) between FRLM and KDEFRLM for a user request (user ID 763) where $T' = \{$`art, city-walks, cafés, fast-food, museums, parks, restaurants, tourism, shopping-for-wine, shopping-for-accessories`$\}$. Both FRLM and KDEFRLM assign higher weights to terms such as ‘park’, ‘museum’, which are clearly relevant for this particular example. Indeed, both these models are also successful at capturing other relevant terms such as ‘view’, ‘tree’, ‘canal’ etc.

[1] `https://github.com/tmikolov/word2vec`

		Graded Evaluation Metrics			Binary Evaluation Metrics			
	Method	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
	IR-based approaches							
BL1	BM25	0.2747	0.2484	0.2889	0.3934	0.3066	0.1326	0.6539
BL2	Term Sel.	0.2484	0.2383	0.3034	0.3639	0.3066	0.1466	0.6148
BL3	BM25 + Term Sel.	0.2411	0.2332	0.3143	0.3672	0.3115	0.1530	0.5607
BL4	RLM [51]	0.2615	0.2453	0.3091	0.3574	0.3033	0.1437	0.6441
BL5	KDERLM [84]	0.2829	0.2682	0.3191	0.3967	0.3361	0.1495	0.6539
	RecSys based approaches							
BL6	Most Popular K	0.1861	0.1926	0.2580	0.2787	0.2705	0.1016	0.4154
BL7	Profile Popular K	0.2488	0.2409	0.2811	0.3410	0.3016	0.1280	0.6486
BL8	NeuMF [45]	0.1626	0.1655	0.2480	0.2361	0.2344	0.0937	0.4314
BL9	Bayesian	0.2170	0.1774	0.1816	0.3082	0.2082	0.0672	0.5831
	Hybrid approaches							
BL10	Content + Tag. [2]	0.2499	0.2411	0.2800	0.3967	0.3377	0.1330	0.5390
BL11	Hybrid (BL5 + BL10)	0.2805	0.2667	0.3329	0.3902	0.3311	0.1583	0.6514
	Proposed approach							
	FRLM ($\gamma_H = 0.8$)	0.2919	0.2810‡	0.3418*†‡	0.3934	0.3443‡	0.1616*‡	**0.6786**
	KDEFRLM ($\gamma_H = 0.6$)	**0.2996**†‡	**0.2868**†‡	**0.3490***†‡	**0.4295**†‡	**0.3656**†‡	**0.1725***†‡	0.6553

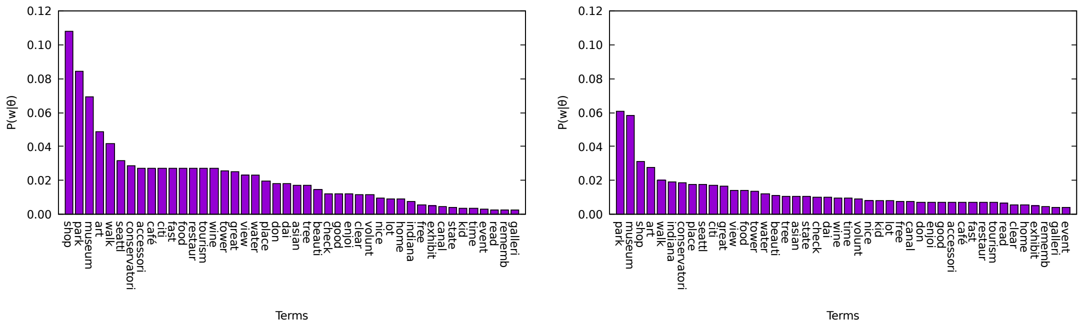
Table 5.1: Comparisons between POI retrieval approaches with location only (*hard*) constraint. The notations, ‘*’, ‘†’ and ‘‡’ denote significant (paired t-test with 95% confidence) improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid), and BL1 (BM25), respectively.

5.5.2 Incorporating Term Semantics Improves POI Effectiveness

We observe from Table 5.1 that the KDE extended version of the factored model mostly outperform its non-semantic (non-KDE) counterpart. This shows that leveraging underlying term semantics of a collection in the form of an embedded space of vectors helps to retrieve more relevant POIs at better ranks. Figure 5.2 shows that KDEFRLM is able to capture the semantic relationship between terms better than FRLM. For example, the semantic relationship between the term ‘histori’ (stemmed form of ‘history’) and ‘museum’ was successfully captured by the KDE-based variant of FRLM. This demonstrates that KDEFRLM is able to successfully leverage the semantic association between terms, in addition to those of the term-based statistical co-occurrences only. Table 5.2 shows a few terms whose word vectors are in close proximity of the user assigned tags in the embedded space.

Tags	Semantically close terms
`beer`	`tap`, `draft`, `craft`, `microbrew`, `draught`, `ipa`, `pint`, `breweri`, `hefeweizen`, `delirium`, `lager`
`beach`	`oceanfont`, `ocean`, `lifeguard`, `pier`, `beachfront`, `sand`, `surfer`, `pismo`, `murrel`, `seasid`, `vacat`
`seafood`	`shellfish`, `oyster`, `crab`, `fish`, `shrimp`, `triggerfish`, `restaur`, `fisherman`, `swordfish`, `lobster`, `scallop`
`pub`	`irish`, `gastropub`, `bar`, `fado`, `behan`, `sport`, `british`, `linkster`, `mccool`, `mcgregor`, `alehous`
`family`	`oper`, `kid`, `pantuso`, `parent`, `niec`, `children`, `orient`, `sicilli`, `fun`, `home`, `yohan`

Table 5.2: (Stemmed) words whose vectors are close to the user assigned tags in the (`word2vec`) embedded space.



(a) FRLM

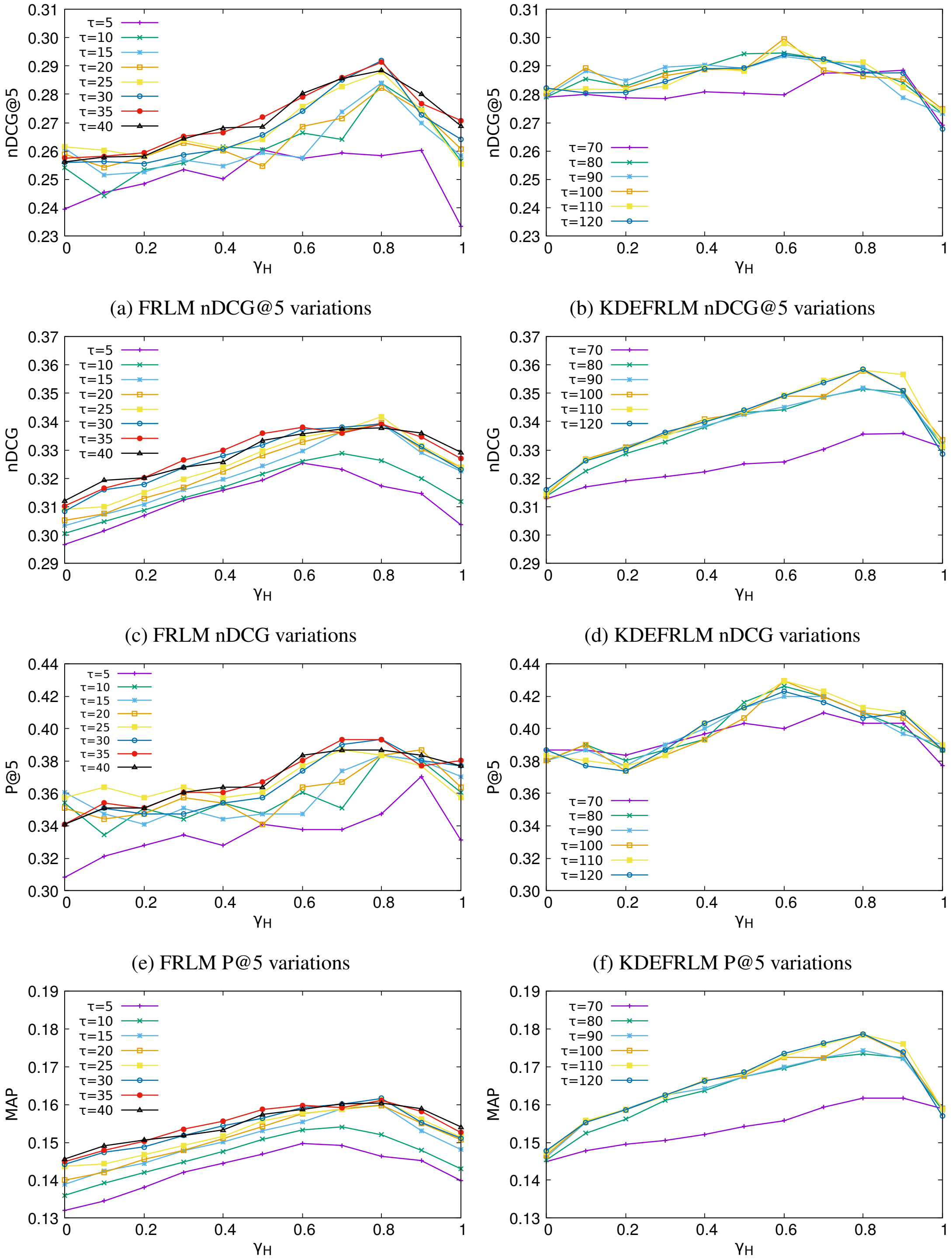
(b) KDEFRLM

Figure 5.2: Comparisons of term distribution weights (sorted from highest to lowest) between FRLM and KDEFRLM on single (location) context. For location-only modeling, FRLM uses $M = 5$ (number of top-retrieved documents for feedback as per the $M(\theta_U, q_U, l_U)$ notation of Table 3.1) and $\tau = 25$ (number of top-scoring terms in the estimated RLM distributions). KDEFRLM uses $M = 2$ and $\tau = 80$.

5.5.3 KDEFRLM Sensitivity

In this section, we investigate the effects of varying the parameter γ_H (i.e. the trade-off between exploration and exploitation) on the performance of KDEFRLM. Figure 5.3 shows the comparative sensitivity of FRLM versus KDEFRLM (measured with nDCG@5, nDCG, P@5 and MAP) with respect to the number of terms (τ) used to define the weighted distribution of terms, and the relative importance of the historical context of a user with respect to the POIs in the current context, i.e. γ_H.

Similar to FRLM, KDEFRLM performs best with a balanced trade-off between exploration and exploitation. In particular, the optimal results (in terms of nDCG@5) is obtained when $\gamma_H = 0.6$. Moreover, the effectiveness of KDEFRLM is not good with the user profile history only, which indicates the diverse



(a) FRLM nDCG@5 variations

(b) KDEFRLM nDCG@5 variations

(c) FRLM nDCG variations

(d) KDEFRLM nDCG variations

(e) FRLM P@5 variations

(f) KDEFRLM P@5 variations

(g) FRLM MAP variations

(h) KDEFRLM MAP variations

Figure 5.3: Effect of precision at top ranks (nDCG@5 and P@5) and recall (nDCG and MAP) with respect to changes in number of terms used in FRLM (location only context) Vs KDEFRLM (location only context) estimation (τ) and the relative importance assigned to user profile information (γ_H).

nature of the historical information itself and demonstrates the usefulness of selectively extracting pieces of information from the history that are contextually relevant in the current situation.

KDEFRLM also shows a similar trend in the selection of terms. We observe that too few or too large a number of terms tends to decrease retrieval effectiveness. The former is not able to adequately capture the relevant semantics required to match the profile with the current context, while the latter introduces noise (from parts of profile that are not relevant in the current context) in the estimated KDEFRLM distribution. Interestingly while FRLM achieves the optimal results with a smaller number of expansion terms, τ, KDEFRLM being a more complex model requires a larger number of expansion terms to perform well.

5.6 Summary

Our experiments (Chapter 4) show that the initial version of the proposed factored relevance model (FRLM) is effective in matching the content between the POIs in user's preference history and the POIs in the current context by estimating a term weight distribution from both information sources. However, the user profile based RLMs as presented in Chapter 4 (θ_{U,q_U} of Equation 4.2 or its factored version, θ_{U,q_U,l_U}, of Equation 4.3) can take into account only the document level co-occurrence of terms (ignoring any semantic associations between them). In fact, improvements in the effectiveness of FRLM was not significant specifically with respect to precision at top ranks (nDCG@5, P@5).

Hence to further improve the retrieval effectiveness at top ranks, we incorporate term semantic information into the FRLM in the form of word vector similarities, and propose a word embedding based (estimated with kernel density estimation) further generalized version of factored relevance model, KDE-FRLM. This leads to a better semantic match between the POI descriptions and the review/description text of the locations visited in the past by a user, which eventually achieves significantly better retrieval performance.

Chapter 6

Multi-Contextual Appropriateness in POI Recommendation

Until this point our proposed models, the factored relevance model (FRLM) and its KDE based variant, have been able only to address the location (*hard*) constraint in POI recommendation. In this chapter, we propose a multi-contextual extension to our proposed models so as to additionally take into account a set of *soft* (trip-qualifier) constraints.

6.1 Weakly Supervised Approach for Addressing Trip Qualifier (*Soft*) Constraints

To incorporate non-location type qualifiers, one needs to learn an association between a word from the review text or the tag vocabulary of a user profile, and the likely (historical) context (trip-type, duration, etc.) that leads to creating the review text in the first place. As an example, it should be possible for humans (with their *existing knowledge*) to infer that a review about a pub frequently mentioning phrases, such as ‘friends’, ‘good times’, ‘tequila shots’ etc. is most likely associated with accompaniment by friends on vacation (i.e. `trip-type=vacation` and `accompanied-by=friends`).

A computational approach to automatically constructing this association requires the use of a knowledge base (e.g. a seed set of term-category associations). One such knowledge resource was compiled in [3], which is composed of the following two different types of manually assessed information.

1. List of pairs constituting a term and a *single* non-location trip-qualifier with manually judged relevance scores of the form (t, q, a), where t is a term (e.g. food), q is a single category (e.g. holiday) and $a \in [0, 1]$) is a manually judged *appropriateness score*. An example of a non-relevant pair is (`nightlife, business, 0.1`) with a lower score. Table 6.2 shows more examples of this sort.
2. List of pairs of a term with a *joint* context (a 3-dimensional vector of categories) along with a man-

Categories	Values
Q_1: `trip-type`	{`business`, `holiday`, `other`}
Q_2: `trip-duration`	{`day-trip`, `longer`, `night-out`, `weekend-trip`}
Q_3: `accompanied-by`	{`alone`, `family`, `friends`, `other`}

Table 6.1: Soft constraint categories with their values (this table is reproduced from Chapter 3 for the sake of convenience).

#Assessors	Appropriateness	Term/Phrase	Single Context (Q_i)
12	1.00	American Restaurant	`trip-duration=weekend-trip`
7	0.71	American Restaurant	`trip-duration=longer`
12	-0.48	Nightlife Spot	`trip-type=business`
7	-1.0	Nightlife Spot	`accompanied-by=family`

Table 6.2: Crowd sourced contextual appropriateness data for *single context* [3]. Table 6.1 lists the categorical values corresponding to the three trip qualifiers (this table is reproduced from Chapter 3 for the sake of convenience).

#Assessors	Appropriateness	Term/Phrase	$Q = Q_1 \times Q_2 \times Q_3$ (`trip-type`, `trip-duration`, `accompanied-by`)
3	1.0	Movie Theater	‘`holiday, day-trip, friends`’
3	1.0	Irish Pub	‘`holiday, night-out, friends`’
3	1.0	Steakhouse	‘`business, longer, family`’
3	-1.0	Bar	‘`holiday, weekend-trip, family`’
3	1.0	Bar	‘`holiday, weekend-trip, alone`’
3	-1.0	Grocery Store	‘`business, day-trip, alone`’

Table 6.3: Crowd-sourced contextual appropriateness data for *joint context* [3] (this table is reproduced from Chapter 3 for the sake of convenience).

ually assessed binary label (1/0) indicating whether the term is relevant in the given *joint context* or not. As an example, the word ‘pub’ is assessed to be non-relevant in the joint context of ‘(`holiday, family, weekend`)’, whereas it is relevant in the context ‘(`holiday, friends, weekend`)’. Table 6.3 shows more examples of this sort.

We formally denote these two types of knowledge resources (Tables 6.2 and 6.3) as

$$\begin{aligned} \kappa_s &: (w, q) \mapsto [0, 1], w \in V, q \in Q_i, i \in \{1, \ldots, c\} \\ \kappa_j &: (w, q) \mapsto \{0, 1\}, w \in V, q \in Q = Q_1 \times \ldots Q_c, \end{aligned} \tag{6.1}$$

where Q denotes the set of *joint* non-location type contexts (*soft* constraints), Q_i denotes a *single* context category, and V denotes the vocabulary set of the review text and tags.

A seed set of such labeled examples of term-context (single or joint) association pairs can then be used to define a modified similarity score function ψ. In contrast to the text-based function of Equation 3.2, this also takes into account the information from the *soft* constraints of the query context. In particular for a given *soft* constraint vector q_U in the user query, we use embedded word vector representations to aggregate the similarities of each word in the review text/tag of a user profile with the seed words assessed as relevant for a single or a joint context q_U. Formally, $\forall w \in U$ we define two functions of the form $\psi : (w, q_U) \mapsto \mathbb{R}$, one each for the addressing the single and the joint contexts, as shown in Equation 6.2.

$$\begin{aligned} \psi_s(w, q_U) &= \max(\mathbf{w} \cdot \mathbf{s}),\ s \in \cup\{t : \kappa_s(t, q_U) > 0\} \\ \psi_j(w, q_U) &= \max(\mathbf{w} \cdot \mathbf{s}),\ s \in \cup\{t : \kappa_j(t, q_U) = 1\} \end{aligned} \tag{6.2}$$

Equation 6.2 shows that for each word w (embedded vector of which is represented as $\mathbf{w}$) contained in the text from the historical profile of a user, we compute its maximum similarity:

- In the case of single context (ψ_s), over all seed words, and
- In the case of the joint context (ψ_j), over a subset of seed words relevant only for the given context, i.e., the words for which $\kappa(q_U, s) = 1$.

We use `word2vec` [68], to embed the vector representation of a word (similar to the KDEFRLM approach described in Chapter 5).

The reason for using the maximum as the aggregate function in Equation 6.2 is that a word is usually semantically similar to a small number of seed words relevant to a given context. To illustrate this with an example, for the query context 'holiday, day-trip, friends', the relevant seed set constitutes words such as 'base-ball stadium', 'beer-garden', 'salon', 'sporting-goods-shop', etc. However, a word such as 'pub' is similar to only one member of this seed set, namely 'beer-garden', which means that other aggregation functions, such as averaging, can lead to a low aggregated value, which is not desirable in this case.

6.2 Factored Relevance Model with *Soft* Constraints

To incorporate the multi-contextual appropriateness measure into our proposed factored relevance model (FRLM), we combine both the text-based similarity ϕ (Equation 3.2), and the trip context driven similarity function ψ (ψ_s or ψ_j of Equation 6.2) into our proposed relevance models. Specifically, the user profile based RLM of Equation 4.2 is generalized as shown in Equation 6.3.

$$P(w|\theta_{U,q_U}) = \sum_{(D,T,r)\in U} rP(w|D)\psi(w,q_U) \prod_{t\in T'} P(t|D) \tag{6.3}$$

In addition to addressing the semantic relationship between a user assigned tag and a term present in the POI description, this relevance model of Equation 6.3 also takes into account the trip-qualifier based contextual appropriateness of a term w by the use of the $\psi(w,q_U)$ factor. A higher value of this factor indicates that either w is itself one of the seed words in an existing knowledge base or its embedded vector is close to one of the seed words, thus indicating its likely contextual appropriateness. It is worth noting that substituting an identity function for $\psi(w,q_U)$, i.e., $\psi_l : (w,q) \mapsto 1$, degenerates the general case to the particular case of *location-only* user-profile based RLM of Equation 4.2.

In a similar manner, the *exploration* part of the model (Equation 4.3) is generalized as shown in Equation 6.4.

$$P(w|\theta_{U,q_U,l_U}) = \sum_{d\in M(\theta_U,q_U,l_U)} P(w|d)\psi(w,q_U) \prod_{t\in\theta_{U,q_U}} P(t|d) \tag{6.4}$$

More specifically, the *soft* constraint similarity function ψ manifests itself in three different forms, namely $\{\psi_l, \psi_s, \psi_j\}$, for the location (*hard* constraint) only retrieval, single-context based retrieval and joint-context based retrieval, respectively.

The word-semantics enriched relevance models (KDEFRLM) can also be generalized by incorporating the ψ function within them to further generalize them to address multiple contexts. Similar to the non-semantic version of the factored relevance model, the multi-contextual appropriateness measure, $\psi(w,q_U)$, is incorporated into the KDE based FRLM model as a part of the kernel function weights $\alpha_t = P(w|\mathcal{M})\psi(w,q_U)P(t|\mathcal{M})$ in Equation 5.5, as shown in Equation 6.5.

$$P(w|\theta_{U,q_U};h,\sigma) = \sum_{t\in T'} \Big(\sum_{(D,T,r)\in U} rP(w|D)\Big)\psi(w,q_U)P(t|\mathcal{M})\frac{1}{\sigma\sqrt{2\pi}}exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2h^2}) \tag{6.5}$$

Finally, the *exploration* side of the model is generalized as shown in Equation 6.6.

$$P(w|\theta_{U,q_U,l_U};h,\sigma) = \sum_{d\in M(\theta_U,q_U,l_U)} \frac{1}{\sigma\sqrt{2\pi}}P(w|d) \prod_{t\in\theta_{U,q_U}} P(t|d)\psi(w,q_U)\exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2h^2}) \tag{6.6}$$

Equation 6.6 is the most general among our proposed family of models, the contributing factors being

1. θ_{U,q_U}, which takes into account an enriched user profile while matching against POIs of the current location,
2. $\exp(-\frac{(\mathbf{w}-\mathbf{t})^T(\mathbf{w}-\mathbf{t})}{2\sigma^2h^2})$, which addresses the semantic association between tags and document terms (both user profile and POI descriptors of the current location), and
3. $\psi(w,q_U)$, which factors in the trip-qualifier based contextual appropriateness.

6.3 Methods Investigated

Following the experimental setup described in Section 3.2 (Chapter 3), we employ the same set of IR-based, recommender system (RecSys) based, and hybrid methodologies as baselines, as we did for FRLM in Chapter 4, and KDEFRLM in Chapter 5, for comparison against our generalized model. In particular, **BM25**, **Term Selection**, (**BM25 + Term Selection**), **RLM** [51], and **KDERLM** [84] have been employed as IR-based baselines. On the other hand, we employ **Most Popular K**, **Profile Popular K**, **NeuMF** [85] as our RecSys baselines. Hybrid approaches include **Content + Tag** [2, 1], and a **hybrid** of KDERLM, and Content + Tag.

Generally speaking, in addition to investigating the overall effectiveness of alternative approaches, with respect to our proposed models, we explore the following.

- Finding an optimal trade-off between a user's preference history (*exploitation*) and the information about the POIs constrained to a *hard* contextual constraint such as 'location' (*exploration*) for contextual POI recommendation.
- Finding the most effective way to include *soft* contextual constraints such as 'trip-type', 'accompanied-by' of a given user profile into the POI recommendation framework with a particular focus to improve the precision at top ranks.

With respect to the second objective above, the choice of the *soft* constraint similarity function $\psi = \{\psi_l, \psi_s, \psi_j\}$ yields three different versions for each method investigated, corresponding to: i) not using the soft constraints (i.e. location-only based retrieval), ii) using the single-context, iii) using the joint-context based similarities, respectively. In our results reported in Table 6.4, 6.5, and 6.6, we denote this choice of our model instantiation by an additional parameter for the function ψ. The function corresponding to only location (*hard*) constraints corresponds to the constant function $\psi_l : (w, q) \mapsto \{1\}$. To enable fair comparisons of standard baselines with the proposed models, we extend standard baseline approaches with the *soft* constraints as well, which we describe next.

In BM25, for each query term t, we include the value of $\psi(t, q_U)$ as the weight of that term in the query. Similar to BM25, in case of Term Selection, we include the value of $\psi(t, q_U)$ as the weight of each selected term t, in the query, which remains consistent for BM25 + Term Selection.

For the baseline approach RLM, we use traditional 'RM3'. To incorporate the *soft* contextual constraints into the traditional RLM framework, we include the weights obtained from the ψ function (external knowledge resource) as weights into the standard RLM equation (Equation 4.1). Similarly, the *soft* contextual constraints are incorporated within KDERLM (Equation 5.2) as weighting factors computed with the ψ function.

Soft constraint based variants of the baselines Most Popular K, Profile Popular K, and NeuMF include the value of $\psi(t, q_U)$ as the weight of each selected tag/term t in a query. Since **Bayesian** is primarily a text classification based approach, and there is no direct notion of weighted query with varying term importance, we limit use of this baseline to our *hard* constraint only experiments.

Previous research [2, 1] investigated the use of separately computing a similarity score between the query words/tags and document (POI) words/tags (**Content + Tag** score) with a predicted likelihood score of the relevance between a query word and a given non-location (*soft*) constraint category. As per [2, 1], we trained an SVM-based binary classifier on the joint-context knowledge resource [3] (with relevance labels 0/1) using as inputs the scores for the single contexts. While testing (i.e., at query time), the distance of a 3-dimensional joint context input from the classifier boundary is added to the text (tag-word) matched score (higher the distance, the higher is the likelihood of a tag to be appropriate to the given joint context). We employ this approach as a baseline and denote it by '**Content + Tag + SVM**'. Additionally, we also investigate the method of adding the scores obtained from the ψ_s and ψ_j functions in conjunction with the 'Content + Tag' approach.

Parameters for each method were separately tuned with the help of a grid search. As mentioned earlier for FRLM in Chapter 4, since our proposed models are unsupervised (without involving any parameter learning with the help of gradient descent updates), we do not employ a separate train and test split for conducting grid search. Two common parameters to all the relevance feedback models are the number of feedback documents, M, and the number of feedback terms, τ. It was found after a grid search that RLM and FRLM yielded optimal results with the values 5 (#documents) and 25 (#terms). Similarly for KDERLM, M and τ were optimized to the values 3 and 80, whereas for KDEFRLM, the optimal values of M and τ were found to be 2 and 100, respectively. As mentioned earlier in Section 3.2 (Chapter 3), for our experimental setup, as shown in Equation 6.1, we normalized the contextual appropriateness scores for both single and joint context, within $[0, 1]$.

6.4 Word Embedding Settings

In this section, we discuss about word embedding setup which is required for our embedding based model KDEFRLM, and in modeling multiple *soft* contextual constraints for both FRLM and KDEFRLM. Since different choices in an embedding method, such as the embedding objective function or the collection on which the embedding model is trained on etc., may influence the retrieval effectiveness [83], we explore four different ways for generating the embedded word vectors. In fact, we report the performance variation of our proposed model KDEFRLM as obtained with a number of different embedding methodologies in Table 6.7.

Distances between word vectors are used in the kernel density based approaches and in modeling the soft constraints. Specifically, for our experiments (Table 6.5) the embedded space of word vectors is obtained by executing skipgram [68] with default values for the parameters of window-size (5) and the number of negative samples (5), as set in the word2vec tool[1]. Skipgram was trained on the collection of the POI descriptors in the TREC-CS collection. We mention this version of word embeddings as word2vec-In i.e. *In-domain* (Table 6.7) as it is trained on the target corpus.

As an alternative to training word vectors on the target collection, we also explore pre-trained word vectors trained on large external corpora, which is a common practice for supervised NLP downstream tasks [83, 31, 98]. Specifically, we employ two word embedding methodologies word2vec (Out-domain), and GloVe (Out-domain) [74]. We also employ BERT (Out-domain) [31] which is a contextual embedding method that uses masked language models.

While the 300 dimensional word2vec pre-trained vectors that we used were trained on Google news dataset[2], the 300 dimensional GloVe vectors that we used were trained on the Common Crawl[3]. The transformer based contextual vectors that we used for our experiments uses the pre-trained RoBERTa [60] model, which is an optimized version of the original BERT model. Given a word, the RoBERTa model outputs a 768 dimensional vector. Since the objective of this set of experiments is to investigate the effect of different embedding approaches on the effectiveness of our proposed model, the remaining parameters for KDEFRLM method such as the number of feedback documents, M, and the number of expansion terms, τ, were set to their optimal values as tuned on the word2vec-In experiments.

6.5 Results

Table 6.4, 6.5, and 6.6 show the results obtained by each contextual recommendation approach that we investigated, as outlined in Section 6.3. We present a summary of the optimal results for both the location-based and the location + trip-qualifier based approaches in three tables - 1) comparison between our proposed models and IR-based approaches, 2) proposed models versus RecSys-based approaches, and 3) proposed models versus hybrid approaches. Each method was separately optimized with grid search on the nDCG@5 metric, the official metric to rank systems in the TREC-CS task.

Since the effectiveness of a particular approach (e.g. FRLM) in comparison to a baseline (e.g. RLM) is comparable across the same setting (i.e., location-only or location + soft constraints), we present the comparable rows in separate colour codes (light-grey for the location-only results, i.e., ψ_l, and no colour for the soft constraints based results, i.e., ψ_s and ψ_j,) so that only the rows with the same colour code are comparable to each other.

We first report the results of our set of experiments and summarize the overall observations. Then we investigate the sensitivity analysis of our models with different contextual constraint settings. Finally, we discuss about the effect of different embedding techniques on the effectiveness of our proposed model.

[1] https://github.com/tmikolov/word2vec

[2] Available at https://code.google.com/archive/p/word2vec/

[3] Available at https://nlp.stanford.edu/projects/glove/

	Context	Graded Evaluation Metrics			Binary Evaluation Metrics			
Method	(ψ)	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
IR-based approaches								
BL1 BM25	ψ_l	0.2747	0.2484	0.2889	0.3934	0.3066	0.1326	0.6539
BL1 BM25	ψ_s	0.2609	0.2441	0.2889	0.3869	0.3164	0.1335	0.5967
	ψ_j	0.2641	0.2464	0.2916	0.3639	0.3033	0.1355	0.6565
BL2 Term Sel.	ψ_l	0.2484	0.2383	0.3034	0.3639	0.3066	0.1466	0.6148
BL2 Term Sel.	ψ_s	0.2424	0.2411	0.3039	0.3607	0.3148	0.1458	0.6186
	ψ_j	0.2539	0.2447	0.3099	0.3705	0.3197	0.1514	0.6419
BL3 BM25 + Term Sel.	ψ_l	0.2411	0.2332	0.3143	0.3672	0.3115	0.1530	0.5607
BL3 BM25 + Term Sel.	ψ_s	0.2462	0.2471	0.3207	0.3672	0.3344	0.1578	0.6095
	ψ_j	0.2530	0.2429	0.3195	0.3869	0.3328	0.1557	0.6191
BL4 RLM [51]	ψ_l	0.2615	0.2453	0.3091	0.3574	0.3033	0.1437	0.6441
BL4 RLM [51]	ψ_s	0.2583	0.2466	0.3107	0.3475	0.3016	0.1443	0.6441
	ψ_j	0.2692	0.2514	0.3189	0.3639	0.3131	0.1496	0.6544
BL5 KDERLM [84]	ψ_l	0.2829	0.2682	0.3191	0.3967	0.3361	0.1495	0.6539
BL5 KDERLM [84]	ψ_s	0.2839	0.2668	0.3236	0.3902	0.3902	0.1530	0.6639
	ψ_j	0.2772	0.2666	0.3287	0.3869	0.3311	0.1594	0.6623
Proposed approaches								
FRLM ($\gamma_H = 0.8$)	ψ_l	0.2919	0.2810‡	0.3418*†‡	0.3934	0.3443‡	0.1616*‡	**0.6786**
FRLM ($\gamma_H = 0.8$)	ψ_s	0.2956	0.2806	0.3435	0.4033	0.3443	0.1637	0.6922
	ψ_j	0.3075	0.2935	0.3498	0.4098	0.3541	0.1687	0.7098
KDEFRLM ($\gamma_H = 0.6$)	ψ_l	**0.2996**†‡	**0.2868**†‡	**0.3490***†‡	**0.4295**†‡	**0.3656**†‡	**0.1725***†	0.6553
KDEFRLM ($\gamma_H = 0.7$)	ψ_s	0.3079	0.2852	0.3502	0.4361	0.3557	0.1729	0.6648
	ψ_j	**0.3199***†‡	**0.2980***†‡	**0.3645***†‡	**0.4426***†‡	**0.3623***†‡	**0.1824***†‡	**0.7143**

Table 6.4: Comparisons between proposed models and IR-based approaches. The notations, '$*$', '$\dagger$' and '$\ddagger$' denote significant (paired t-test with 95% confidence) improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25), respectively.

6.5.1 Overall Observations

Some observations that we already mentioned in Chapter 4, 5, such as, the factored models (exploration and exploitation) outperform the other approaches, IR approaches outperform collaborative/personal RecSys ones, unsupervised approaches outperform supervised ones etc. are common here.

	Context	Graded Evaluation Metrics			Binary Evaluation Metrics			
Method	(ψ)	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
RecSys based approaches								
BL6 Most Popular K	ψ_l	0.1861	0.1926	0.2580	0.2787	0.2705	0.1016	0.4154
BL6 Most Popular K	ψ_s	0.1765	0.1894	0.2579	0.2590	0.2689	0.1015	0.4055
	ψ_j	0.1877	0.1844	0.2590	0.2656	0.2475	0.1010	0.4247
BL7 Profile Popular K	ψ_l	0.2488	0.2409	0.2811	0.3410	0.3016	0.1280	0.6486
BL7 Profile Popular K	ψ_s	0.2529	0.2381	0.2861	0.3639	0.3016	0.1321	0.6296
	ψ_j	0.2568	0.2487	0.2908	0.3574	0.3098	0.1362	0.6500
BL8 NeuMF [45]	ψ_l	0.1626	0.1655	0.2480	0.2361	0.2344	0.0937	0.4314
BL8 NeuMF [45]	ψ_s	0.1491	0.1601	0.2466	0.2131	0.2344	0.0935	0.3969
	ψ_j	0.1698	0.1834	0.2457	0.2393	0.2525	0.0923	0.4300
BL9 Bayesian	ψ_l	0.2170	0.1774	0.1816	0.3082	0.2082	0.0672	0.5831
Proposed approaches								
FRLM ($\gamma_H = 0.8$)	ψ_l	0.2919	0.2810‡	0.3418*†‡	0.3934	0.3443‡	0.1616*‡	**0.6786**
FRLM ($\gamma_H = 0.8$)	ψ_s	0.2956	0.2806	0.3435	0.4033	0.3443	0.1637	0.6922
	ψ_j	0.3075	0.2935	0.3498	0.4098	0.3541	0.1687	0.7098
KDEFRLM ($\gamma_H = 0.6$)	ψ_l	**0.2996**†‡	**0.2868**†‡	**0.3490***†‡	**0.4295**†‡	**0.3656**†‡	**0.1725***†	0.6553
KDEFRLM ($\gamma_H = 0.7$)	ψ_s	0.3079	0.2852	0.3502	0.4361	0.3557	0.1729	0.6648
	ψ_j	**0.3199***†‡	**0.2980***†‡	**0.3645***†‡	**0.4426***†‡	**0.3623***†‡	**0.1824***†‡	**0.7143**

Table 6.5: Comparisons between proposed models and RecSys-based approaches. The notations, '$*$', '$\dagger$' and '$\ddagger$' denote significant (paired t-test with 95% confidence) improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25), respectively.

In summary, from Table 6.4, 6.5, and 6.6 we can see that the word semantics based extension of our proposed factored relevance model, i.e., KDEFRLM, outperforms all other methods for both the location-only (hard) and 'location + trip-qualifier' (hard and soft) constrained contextual POI recommendation tasks. A paired t-test showed that the improvements in nDCG@5, nDCG@10, nDCG, P@5, P@10, and MAP with KDEFLRM were statistically significant (95% confidence level) in comparison to the three strongest baselines: BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25). We now highlight and comment on the key observations from our set of experiments.

Figure 6.1 shows the comparison of relative term distributions (common terms) between FRLM and KDE-FRLM for a user request (user ID 763) where $T' = \{$`art, city-walks, cafés, fast-food, museums, parks, restaurants, shopping-for-wine, shopping-for-accessories,`

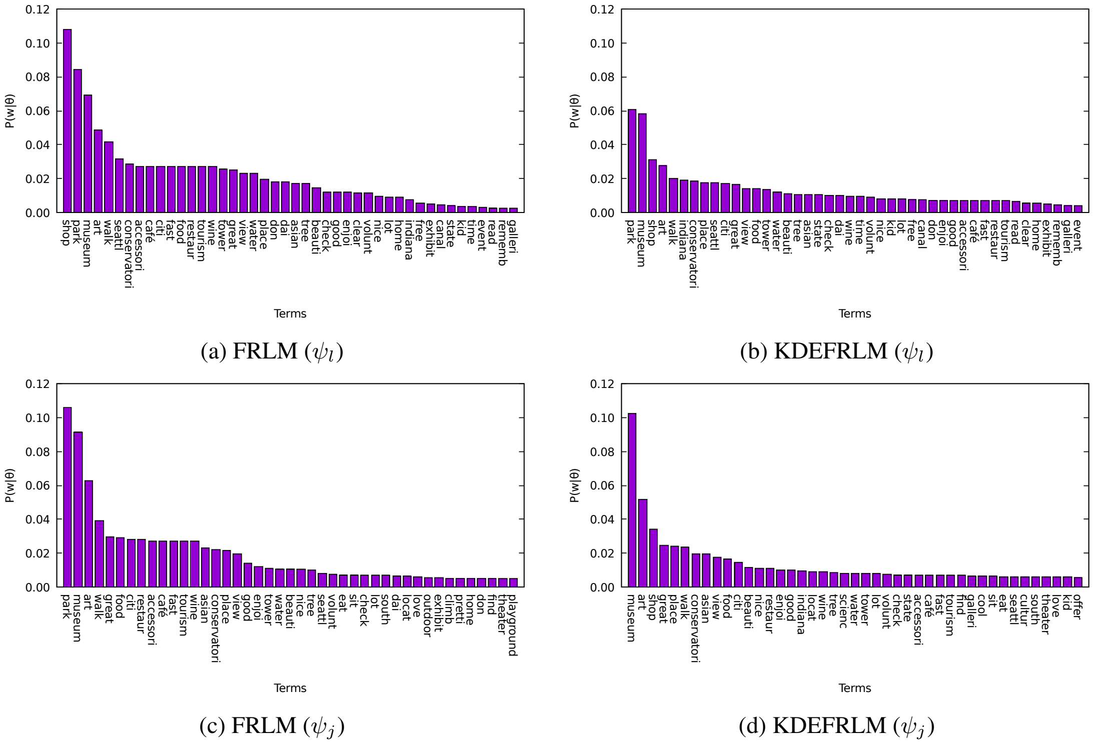
		Context	Graded Evaluation Metrics			Binary Evaluation Metrics			
	Method	(ψ)	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
	Hybrid approaches								
BL10	Content + Tag. [2]	ψ_l	0.2499	0.2411	0.2800	0.3967	0.3377	0.1330	0.5390
BL10	Content + Tag. [2]	ψ_s	0.2623	0.2496	0.2841	0.4066	0.3492	0.1383	0.5982
		ψ_j	0.2688	0.2651	0.2979	0.4000	0.3656	0.1484	0.6260
		SVM	0.2656	0.2476	0.2833	0.3770	0.3262	0.1330	0.5850
BL11	Hybrid (BL5 + BL10)	ψ_l	0.2805	0.2667	0.3329	0.3902	0.3311	0.1583	0.6514
BL11	Hybrid (BL5 + BL10)	ψ_s	0.2777	0.2612	0.3420	0.3869	0.3230	0.1648	0.6540
		ψ_j	0.2771	0.2615	0.3471	0.3902	0.3246	0.1716	0.6586
	Proposed approaches								
	FRLM ($\gamma_H = 0.8$)	ψ_l	0.2919	$0.2810^{\ddagger}$	$0.3418^{*\dagger\ddagger}$	0.3934	$0.3443^{\ddagger}$	$0.1616^{*\ddagger}$	**0.6786**
	FRLM ($\gamma_H = 0.8$)	ψ_s	0.2956	0.2806	0.3435	0.4033	0.3443	0.1637	0.6922
		ψ_j	0.3075	0.2935	0.3498	0.4098	0.3541	0.1687	0.7098
	KDEFRLM ($\gamma_H = 0.6$)	ψ_l	**0.2996**†‡	**0.2868**†‡	**0.3490***†‡	**0.4295**†‡	**0.3656**†‡	**0.1725***†	0.6553
	KDEFRLM ($\gamma_H = 0.7$)	ψ_s	0.3079	0.2852	0.3502	0.4361	0.3557	0.1729	0.6648
		ψ_j	**0.3199***†‡	**0.2980***†‡	**0.3645***†‡	**0.4426***†‡	**0.3623***†‡	**0.1824***†‡	**0.7143**

Table 6.6: Comparisons between proposed models and hybrid approaches. The notations, '*', '†' and '‡' denote significant (paired t-test with 95% confidence) improvements over the three strongest baselines - BL5 (KDERLM), BL11 (Hybrid) and BL1 (BM25), respectively.

tourism}. In addition to location only retrieval (ψ_l), it also shows the term distributions for *hard* + *soft* constraint based retrieval (ψ_j).

6.5.2 Joint Context Modeling is Better for Modeling Soft Constraints

From Table 6.4, 6.5, and 6.6 we observe that including trip-qualifier based information in the form of joint context (ψ_j) generally improves POI retrieval effectiveness, e.g. improvements are observed for RLM, NeuMF, etc. (compare the results between ψ_j and ψ_l for each method). Standard approaches do not benefit much from the inclusion of the trip-qualifiers in the form of single-context driven scores, a plausible reason for which can be attributed to the fact that relevant single-context matches may not lead to the conjunctive relevance for the joint context. However, including even the single context based similarity scores as part of the query term weights in standard IR and RS (recommender system) approaches tends to improve the recall. E.g. effectiveness measures such as MAP and nDCG mostly improve at the cost of a decrease in nDCG@5 or P@5.



(a) FRLM (ψ_l)

(b) KDEFRLM (ψ_l)

(c) FRLM (ψ_j)

(d) KDEFRLM (ψ_j)

Figure 6.1: Comparisons of term distribution weights (sorted from highest to lowest) between FRLM and KDEFRLM on single (location) and multiple contexts (joint modeling with ψ_j). For location-only modeling, FRLM (ψ_l) uses $M = 5$ (number of top-retrieved documents for feedback as per the $M(\theta_U, q_U, l_U)$ notation of Table 3.1) and $\tau = 25$ (number of top-scoring terms in the estimated RLM distributions). KDEFRLM (ψ_l) uses $M = 2$ and $\tau = 80$. FRLM with joint context (ψ_j) uses parameters $M = 5$ and $\tau = 35$, whereas the results for KDEFRLM with the joint context (ψ_j) were obtained with $(M, \tau) = (2, 100)$.

It can be seen that using soft constraint scores as a part of a model is usually more effective than a simple post-hoc combination of these scores with content matching scores (e.g. the relative improvements in FRLM as compared to that of Popular K or Content + Tag).

Additionally, in contrast to a parametric approach, such as SVM, the proposed similarity function ψ_j (Equation 6.2) works better. This is because supervised approaches typically require large quantities of training data to work well. Moreover, the SVM based approach of [1] did not take into account the semantic similarities between words to estimate the trip-qualifier based appropriateness. It is observed that computing similarities with the embedded word vectors turns out to be more effective.

Finally, it can be observed that the best results are obtained when the joint-context based similarity function is incorporated into the factored models. Incorporating term semantics in combination with the soft constraints (KDEFRLM with joint context modeling, ψ_j) further improves the results.

6.5.3 Better Precision-oriented and Recall-oriented Retrieval

In addition to the aforementioned observations, we also note that KDEFRLM results in the best nDCG@5 value (a precision-oriented metric). This indicates that the model is able to retrieve documents assessed to be most relevant towards the top ranks in comparison to the other baselines. This is particularly beneficial from a user satisfaction point-of-view because a user does not need to scroll-down a list of retrieved suggestions to find her likely best matches. It is particularly worth noting the considerable improvements in the nDCG values (which is both a precision and a recall oriented measure) obtained with KDEFRLM. This indicates that KDEFRLM achieves high recall, in addition to achieving high precision. The high recall implies that, in real-life situations, it is also beneficial for patient users who are prepared to explore a list of recommendations to find a set of likely matching venues.

6.5.4 Sensitivity Analysis

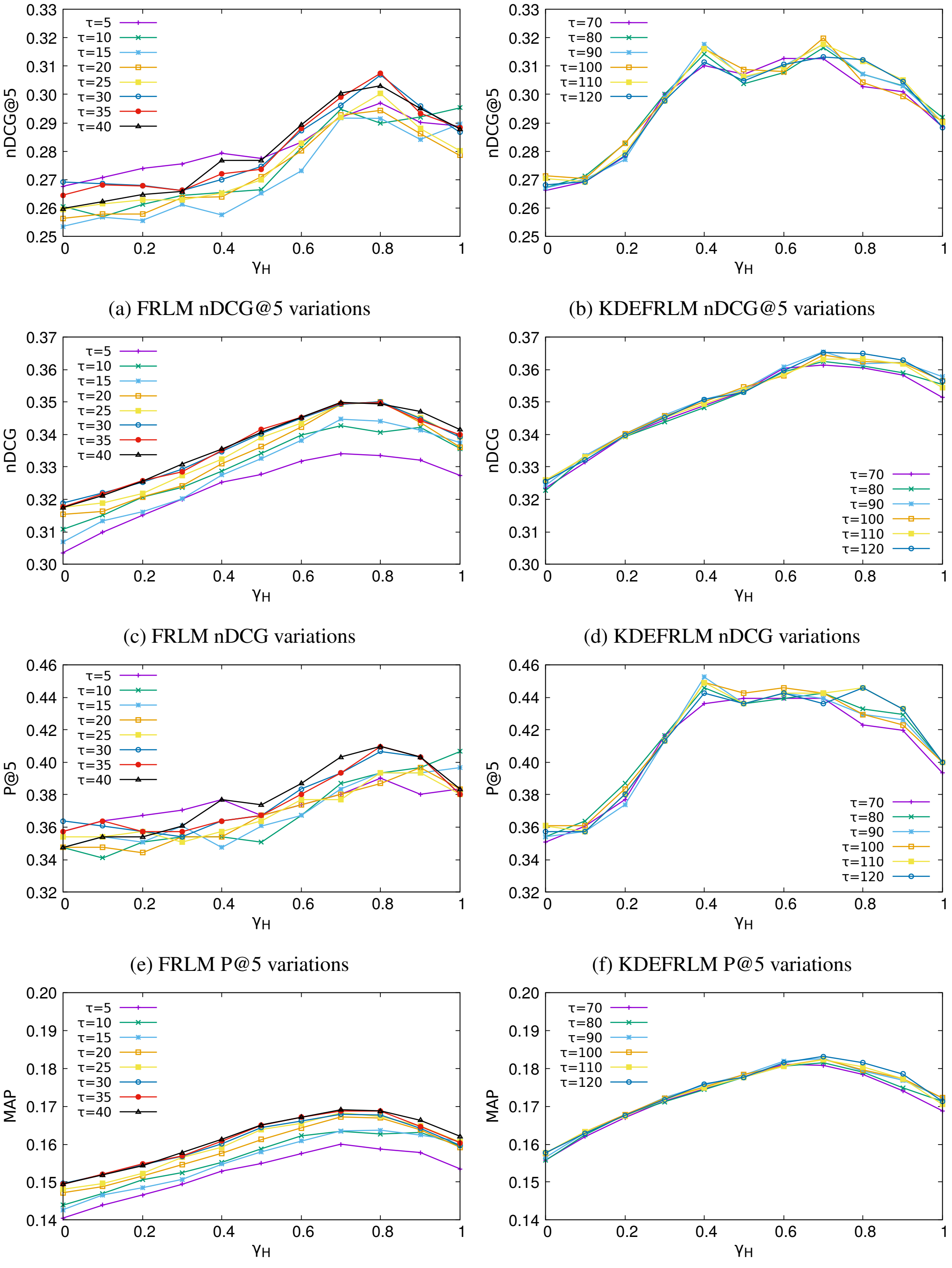
Parameter Sensitivity

Table 6.4, 6.5, and 6.6 present a summary of the best results obtained with each method (parameters optimized with grid-search for the nDCG@5 metric). In order to investigate a more wide spectrum of results, we now investigate the effects of varying the parameter γ_H (i.e. the trade-off between exploration and exploitation) on the performance of FRLM and KDEFRLM. To obtain the sensitivity results, we set the value of M (number of top-retrieved documents to consider for the RLM feedback) to 5, and 2 for FRLM, and KDEFRLM, respectively.

Following a similar trend as shown in Figure 4.3, Figure 6.2 shows the sensitivity of FRLM versus KDEFRLM (measured with nDCG@5, nDCG, P@5 and MAP) with respect to the number of feedback terms, τ, used to define the term-weight distribution, and the relative importance of the user's historical context with respect to the POIs in the current context, i.e. γ_H.

Common observations include the fact that factored relevance models perform best with a balanced trade-off between exploitation and exploration. In particular, the optimal results for FRLM (both in terms of precision oriented measure nDCG@5 and recall oriented measure nDCG) are achieved when $\gamma_H = 0.8$. Moreover, the effectiveness of FRLM degrades with the user profile history only ($\gamma_H = 1$), which indicates that the history information itself is likely to contain noise in the form of topical diversity. This also demonstrates the benefit of selectively extracting chunks of information from the preference history that are contextually appropriate in the present state. We observe a similar trend in the kernel density based extension of FRLM.

It is also observed that a very small or a very large number of feedback terms tends to decrease retrieval performance. While the former case is unable to sufficiently capture the relevant semantics required to match the user profile with the present context, the latter introduces noise from pieces of profile that are not contextually relevant to the present state in the estimated FRLM or KDEFRLM distributions. While FRLM achieves the optimal results with a smaller number of expansion terms, τ, KDEFRLM being a



(a) FRLM nDCG@5 variations

(b) KDEFRLM nDCG@5 variations

(c) FRLM nDCG variations

(d) KDEFRLM nDCG variations

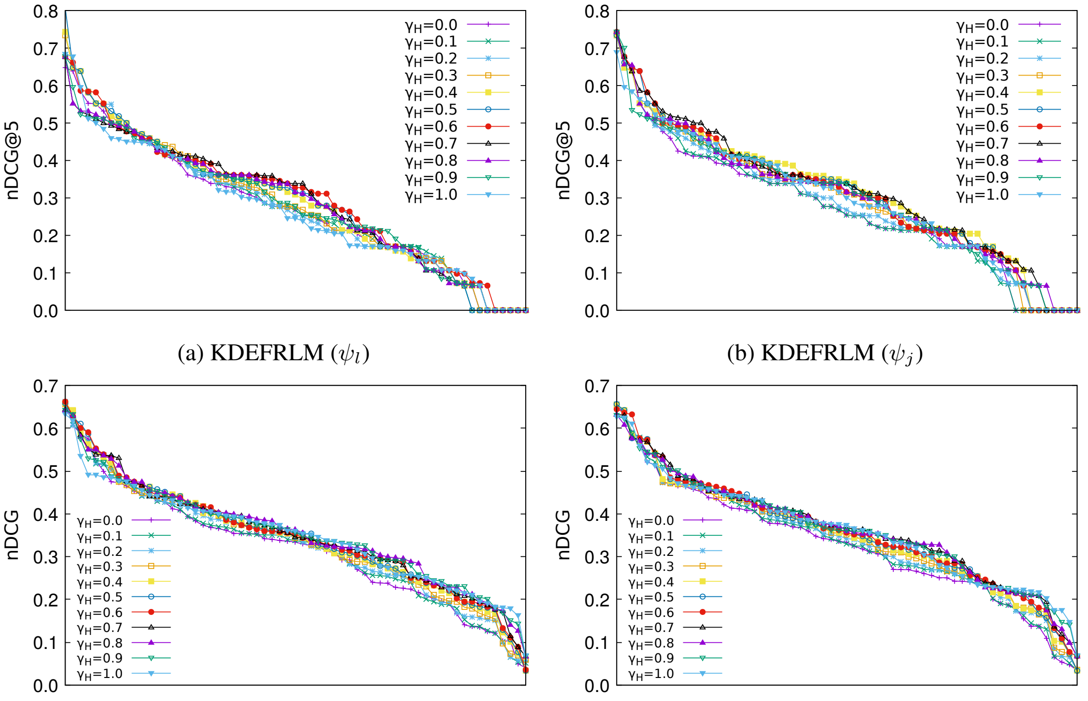
(e) FRLM P@5 variations

(f) KDEFRLM P@5 variations

(g) FRLM MAP variations

(h) KDEFRLM MAP variations

Figure 6.2: Effect of precision at top ranks (nDCG@5 and P@5) and recall (nDCG and MAP) with respect to changes in number of terms used in FRLM (ψ_j) Vs KDEFRLM (ψ_j) estimation (τ) and the relative importance assigned to user profile information (γ_H).



(a) KDEFRLM (ψ_l)

(b) KDEFRLM (ψ_j)

(c) KDEFRLM (ψ_l)

(d) KDEFRLM (ψ_j)

Figure 6.3: User profile based performance of KDEFRLM (ψ_l), and KDEFRLM (ψ_j) with respect to nDCG@5, and nDCG while varying γ_H. User profiles (queries) are sorted in decreasing order of per profile nDCG@5 values. Larger the area under a curve, better the overall performance for that specific γ_H value.

more complex model requires a larger number of expansion terms to perform well. However, KDEFRLM is less sensitive to the number of terms and hence a more robust model as compared to FRLM.

Per User-Profile Sensitivity Analysis

Instead of a relatively simple approach of employing a constant value for the linear combination parameter γ_H, in this section we investigate if individually choosing the values of this parameter based on the user profiles (queries) can lead to better results. In particular, we conduct a grid-based exploration of the parameter γ_H for each query separately.

Figure 6.3 plots the distribution of the nDCG@5, and nDCG values (arranged in a decreasing order) as obtained for a total of 11 possible choices of γ_H for each user profile (query). A larger area under the curve corresponding to a particular value of γ_H indicates that for a higher number of queries this value of γ_H yields optimal retrieval effectiveness.

A large number of cross-over points (as seen from Figure 6.3) of the distribution lines indicates that, generally speaking, different queries achieve optimal results with different values of the exploration-

			Context	Graded Evaluation Metrics			Binary Evaluation Metrics			
Method	Embedding	Domain	(ψ)	nDCG@5	nDCG@10	nDCG	P@5	P@10	MAP	MRR
KDEFRLM	word2vec	In	ψ_l	**0.2996**	**0.2868**	0.3490	0.4295	0.3656	0.1725	**0.6553**
KDEFRLM	word2vec	In	ψ_s	0.3079	0.2852	0.3502	0.4361	0.3557	0.1729	0.6648
			ψ_j	**0.3199**	0.2980	0.3645	**0.4426**	0.3623	0.1824	**0.7143**
KDEFRLM	word2vec	Out	ψ_l	0.2993	0.2840	0.3485	**0.4328**	0.3656	0.1735	0.6367
KDEFRLM	word2vec	Out	ψ_s	0.2990	0.2879	0.3539	0.4393	0.3672	0.1758	0.6424
			ψ_j	0.3044	0.2959	**0.3653**	**0.4426**	0.3787	**0.1844**	0.6582
KDEFRLM	GloVe	Out	ψ_l	0.2963	0.2863	**0.3533**	0.4262	**0.3705**	**0.1750**	0.6451
KDEFRLM	GloVe	Out	ψ_s	0.3107	**0.3027**	0.3623	0.4361	**0.3852**	0.1802	0.6880
			ψ_j	0.3064	0.2959	0.3638	0.4328	0.3754	0.1815	0.6803
KDEFRLM	RoBERTa	Out	ψ_l	0.2971	0.2842	0.3521	**0.4328**	**0.3705**	0.1736	0.6393
KDEFRLM	RoBERTa	Out	ψ_s	0.2904	0.2894	0.3558	0.4230	0.3754	0.1767	0.6506
			ψ_j	0.2957	0.2855	0.3572	0.4295	0.3721	0.1785	0.6423

Table 6.7: Variations in IR effectiveness with respect to different choices of pre-trained (out-domain) embedding vectors in comparison to skipgram trained on the target collection (i.e. the result of Table 6.5 which is reproduced in this table for convenience)

exploitation parameter. This in turn indicates that for some user profiles it is better to rely to a greater degree on the historical preferences (exploitation) whereas for some other ones it is better to allow provision for more exploration into the POI descriptors. Our results also suggests that automatically estimating the value of the exploration-exploitation trade-off can potentially improve results further. This we leave as a future exercise.

6.5.5 Investigating Variations in Embedding Methodology

The KDEFRLM results reported in Table 6.5 used word embeddings trained on the domain specific target collection. In this section, we investigate whether alternative embedding choices (e.g. using a larger and more general external corpora) leads to improvements in results as reported in previous studies [83]. In particular, we investigate three different choices for the embedding algorithm, namely `word2vec` [68], `GloVe` [74] and `RoBERTa` [60], the latter being a context embedding model employing a transformer-based architecture to learn a masked language model.

In the KDEFRLM framework of Equation 6.6, we provide as inputs pre-trained word vectors instead of word vectors trained on the target collection. While the `word2vec` (skipgram) and the `GloVe` vectors are both 300 dimensional (trained respectively on GoogleNews and CommonCrawl), the `RoBERTa` vectors for each word is 768 dimensional. To obtain vector representations of stemmed words we follow the methodology of [83] which involves first partitioning words into equivalence classes of identical stemmed representations and then consider the average vector of each class as the vector representation of the stem. To illustrate with an example, the vector representation of a word, such as 'comput' in the target collection is given by the average over vectors for words in the pre-trained vocabulary, such as 'computing', 'computer' etc.

For location only retrieval (ψ_l), the best retrieval results are obtained (as measured with the precision oriented metrics - nDCG@5 and nDCG@10) with `word2vec` embedding trained on the target (TREC-CS) corpus itself. A likely reason for this is that training on the target collection is possibly able to capture domain specific term semantics in a better way than a generic (domain-agnostic) representation. An interesting observation is that the pre-trained `GloVe` model (trained on an external data of generic web-pages, namely CommonCrawl) leads to KDEFRLM's best performance with respect to the other metrics such as nDCG, P@10, and MAP. One advantage of using pre-trained vectors on external large corpora is that it offers a generalized way of learning word semantics, and may turn out to be effective when the target corpus is not large enough to learn adequate semantic relationships between words.

For multi-contextual retrieval (ψ_s or ψ_j), again the best performance is achieved by KDEFRLM (with respect to the metrics - nDCG@5, and P@5) for the `word2vec` (In-domain) setting. Here, `word2vec` (Out-domain) embedding also turns out to be effective in contributing to the best performance of KDEFRLM with respect to the metrics - nDCG, P@5, and MAP. It turns out that context embedding (specifically `RoBERTa`) is not as effective as the shallow word-level embedding methodologies, specially for the *soft* contextual constraints case. A likely reason for this is that context embeddings have been shown to be particularly suited for downstream NLP tasks [31], and these may not be well suited to model the lexical semantics across words.

6.6 Summary

Trip-qualifiers, such as *trip-type* (vacation, work etc.), *accompanied-by* (e.g., solo, friends, family etc.) are potentially useful sources of information that could be used to improve the effectiveness of POI recommendation in a current context (with a given set of these *soft* constraints). However, using such information is not straight forward because a user's text reviews about the POIs visited in the past do not explicitly contain such annotations (e.g., a positive review about a pub visit does not contain the information on whether the user was with friends or alone, on a business trip or vacation).

In this chapter, we propose to use a small set of manually compiled knowledge resource to predict the associations between the review texts in a user profile and the likely trip contexts. In particular, we propose a word embedding based approach to compute the similarity of a given trip qualifier (part of the query) with a POI description by employing weak supervision from a knowledge resource.

We demonstrate that incorporating this information within an IR-based relevance modeling framework significantly improves POI recommendation. We observed that modeling the trip-qualifier contexts jointly turns out to be the most effective in comparison to not using these qualifiers at all, or modeling them independently.

We would like to mention that both the proposed FRLM and its word embedding based variant (KDE-FRLM) have been developed for both the location only (i.e. *hard* context based) retrieval, and the multi-contextual (i.e. *hard+soft*) recommendation. In addition, we also investigate the choice of *different word embedding techniques (in-domain vs. externally trained)* in the effectiveness obtained with our embedding based model KDEFRLM (in both the kernel density estimation process and also in modeling the *soft* contextual constraints).

Chapter 7

Relevance Feedback with Query Variants

Until this point of the thesis, we have shown the effectiveness of IR-based approaches for contextual POI recommendation. In particular, our proposed approaches (Chapter 4, 5, and 6) are based on a pseudo-relevance feedback (Section 2.1.2, Chapter 2) framework. In this chapter we discuss about the concept of a weakly supervised relevance feedback approach to improve the information retrieval effectiveness in general, which is eventually applied in the specific task of contextual POI recommendation.

To mitigate the problem of over-dependence of a pseudo-relevance feedback algorithm on the top-M document set, we make use of a set of equivalence classes of queries rather than one single query. These query equivalents are automatically constructed either from a) a knowledge base of prior distributions of terms with respect to the given query terms, or b) iteratively generated from a relevance model of term distributions in the absence of such priors. These query variants are then used to estimate the retrievability of each document with the hypothesis that documents that are more likely to be retrieved at top-ranks for a larger number of these query variants are more likely to be effective for relevance feedback. Results of our experiments show that our proposed method is able to achieve substantially better precision at top-ranks (e.g. higher nDCG@5 and P@5 values) for ad-hoc IR and points-of-interest (POI) recommendation tasks.

Contextual POI recommendation is essentially a precision oriented task [43, 1]. Note that the primary objective of this part of work is to improve precision at top ranks for IR methods, which is very important from the user's satisfaction perspective, specifically for POI recommendation problem, as users are often impatient to scroll down the recommendation list. In addition, real-life use-case often requires that results are to be displayed on mobile devices with limited UI resources.

7.1 Introduction

Standard pseudo-relevance feedback (PRF) methods, such as the relevance model and its variants, have in general been shown to improve overall retrieval effectiveness, such as mean average precision. However, these relevance feedback methods can sometimes, at the cost of increasing recall, lead to decreasing the precision at the very top ranks (e.g. for ranks up to 5). This mainly happens because the only source of information which is made available to a PRF method is the top-retrieved set of documents retrieved in response to a query. One of the limitations is that the effectiveness of the PRF algorithms depends, to a large extent, on the choice of this set (the top-retrieved M documents), which makes these algorithms less robust and more sensitive to the variations in the chosen set of pseudo-relevant set of documents [15, 91].

Researchers have explored different approaches to increase the overall retrieval performance, e.g., by learning the appropriate number of feedback terms for query expansion [72], or by selectively using effective feedback terms either by supervised [16] or learning an optimal policy for feedback term selection using reinforcement learning [70]. It was reported in [15] that despite an average performance increase over a set of topics, relevance feedback does not perform well on a large number of topics. One major problem with relevance feedback is that a large number of top ranked (pseudo-relevant) documents may not truly be related to the core information need of the query thus leading to a detrimental effect on the retrieval effectiveness for a large number of topics after query expansion. The study [91] argues that some relevant documents may also in fact act as *poison pills* and hurt post-feedback effectiveness specially in terms of precision.

Our work in this chapter aligns with the approaches that seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents from the top-ranked ones to include in the feedback set. Instances of such work include [52], which uses overlapping clusters of documents to find a number of *dominant clusters* of documents, and [44], which uses a classification approach to decide which documents to include in the feedback set. A key novelty of our work with respect to the existing thread of work for feedback document selection is that *our approach does not rely on one single query for estimating this selector function*. Specifically, our PRF algorithm makes use of an automatically constructed equivalence class of queries instead of a single query, and then uses the query variants to execute multiple retrieval steps. We then leverage the notion of retrievability [6] of a document to estimate the likelihood of its usefulness for relevance feedback. We rely on the assumption that if a document is retrieved at high ranks for a higher number of query variants, it is more likely to be relevant to the information need of the original query and hence more likely to be useful for PRF.

As a way to automatically generate query variants, we leverage information from semantic associations between term pairs, which act as weak supervision signals affecting the subsequent feedback step (hence we call our proposed feedback method *weakly supervised relevance model*, or *WSRM* for short). We argue that our feedback approach is particularly expected to work well in situations where these term pairs are available as manually annotated resources (e.g. knowledge bases). In the absence of a knowledge-base (as in ad-hoc IR), we use a local co-occurrence matrix of term pair relations. Specifically, we construct a graph representing words as nodes, the edge weights between nodes reflecting the co-occurrence likelihoods [87]. We conduct a random walk on this graph to generate the query variants. To demonstrate the efficacy of our feedback approach in both these situations (i.e. without and with available knowledge-bases), we apply our feedback algorithm on two different tasks, namely ad-hoc IR and points-of-interest (POI) recommendation, respectively.

7.2 Related Work

A pragmatic approach towards pseudo-relevance feedback (PRF) essentially relies on term level manipulations, e.g., while Ogilvie et. al. [72] for their query expansion method learn the appropriate number of feedback terms, Cao et. al. [16] selectively use good feedback terms for query expansion. Traditional PRF methods, such as Okapi [78], the relevance model (RM) [51] and its variants [84, 23], primarily rely on the set of top-retrieved M documents for the purpose of selecting potential candidate expansion terms. These approaches inevitably fail to perform well for all queries when the initial top retrieved document set is noisy, which eventually degrade the retrieval performance for many topics after query expansion [15]. For term-level manipulations, researchers have also leveraged on semantic matching with embedded vectors to learn retrieval-specific semantic relationships from top documents retrieved with a large number of queries from a query log [99], or to combine the effects of global term semantics within the framework of RM [84].

A comparatively less explored approach towards PRF is the use of document level manipulations with an aim to create a more robust set of feedback documents [52, 9]. Existing research along this thread includes those of [44] where a supervised classification approach was applied for selecting good feedback documents using a number of features, and [52] where a k-NN based resampling method was applied for selecting the *dominant set* of documents for relevance feedback.

Our proposed document selection method is based on document retrievability [6] on *query variants*. Studying query variants recently became popular among researchers. Use of manually created query variants [8] has been shown to yield more consistent retrieval [7, 13] and query performance prediction effectiveness [101]. In a recent work, Lu et al. [61] explored different fusion techniques to combine multiple relevance models estimated on different query variants. They experimented with both manually created query variants (UQV dataset [8]) and query variants automatically created leveraging external resources. Liu et al. [58] conducted a comparative analysis of manual and automatic query variants and reported that they yield comparable retrieval effectiveness. The study [61] showed that manual query variants result in better query performance prediction (QPP) than automatically constructed variants. Benham et al. [14] explored a way of automatically generating query variants with the help of external parallel corpora to mimic the achievable retrieval performance using manually generated query variants.

Generating query variants based on some external resources may not always be feasible due to the dependency on the external data. Instead of relying on the availability of human generated query variants, in our work we propose a method to generate this set of *reference queries* automatically for each query, without the help of any external resources.

7.3 Weakly Supervised Relevance Model

In this section, we introduce the concept of weakly supervised relevance model. We describe how query variants are automatically constructed and how are they eventually used to select the set of feedback documents.

7.3.1 Relevance Model

As we discussed earlier, a well known PRF method relevance model (RM) [51] essentially estimates a term weight distribution $P(w|R) \approx P(w|Q)$, for a given query $Q = \{q_1, \ldots, q_n\}$. It is assumed that $P(w|R)$ also generate the set of terms in the top-M documents $\mathcal{M} = \{D_1, \ldots, D_M\}$, i.e.,

$$P(w|R) \approx P(w|Q) = \sum_{D \in \mathcal{M}} P(w|D) \prod_{q \in Q} P(q|D). \tag{7.1}$$

From Equation 7.1, it is evident that a high $P(w|Q)$ value (RM term weight) results when a term w occurs frequently in a top-retrieved document (large $P(w|D)$ value) in conjunction with the frequent occurrence of a query term $q \in Q$ within D. Again each mention of 'relevance model' or 'RM' in this work is to be interpreted as its more effective mixture model variant, i.e. 'RM3' [48].

7.3.2 Equivalence Classes of Query Variants

Generally speaking, a PRF model in IR, e.g. a relevance model [51], estimates for each non-query term - a relevance score, which is essentially its local co-occurrence likelihood with the query terms (i.e., within the top-M retrieved). The estimation is based only on a single query usually with a small number of terms.

What a standard feedback model lacks, is the process of accumulating evidences over an *extended* set of a larger number of queries, which may lead to a more robust estimation of the relevance weights. In fact, prior work has shown that a combination of feedback models involving a number of query variants improves the retrieval effectiveness corresponding to the underlying information need of the original query [58]. Notably, both pre-combination (combining PRF models) and post-combination (combining the ranked lists from PRF models) work well in practice [61]. A desirable property of this *extended set of query variants*, comprising a multiple number of queries, is that each member of this set should express a similar information need as that expressed in the original query. We call this set the *equivalence class* of query variants.

A way to construct a good representation of the equivalent set of a query is through a controlled study setup, where participants are asked to formulate queries corresponding to a given information need description ('back-story') [8]. It has been found that the query variants obtained this way, i.e., manually under a controlled setup, for standard TREC query sets (specifically, the TREC Robust, and the TREC

2013 and 2014 Web Tracks) are of relatively good quality in that they can be used to yield a more consistent retrieval [7], improved query performance prediction (QPP) [101] and more effective feedback results [58]. Different from existing approaches of using manually constructed query variants, a key component of our proposed methodology involves automatically generate this set of equivalence class of *query variants* for each query.

7.3.3 Automatic Construction of Query Variants

Local Term Co-occurrences

We first compute the local co-occurrence matrix between the terms present in the vocabulary of (say) the top-M retrieved documents [100]. Specifically,

$$P(u, v; Q, \mathcal{M}) = \frac{1}{|\mathcal{M}|} \sum_{D \in \mathcal{M}} P(u|D)P(v|D), \ u, v \in V_M, \tag{7.2}$$

where the usual notation $P(w|D)$ (similar to RM of Equation 7.1) denotes the probability of sampling a term w from a document D (independent of another term), and the set $V_M = \cup_{D \in \mathcal{M}}\{D\}$ denotes the set of unique terms (vocabulary) of the set of top-retrieved documents $\mathcal{M}$. Similar to RM of Equation 7.1, we employ a standard collection smoothing (Jelinek-Mercer) based maximum likelihood estimate for computing the probabilities, i.e.,

$$P(u|D) = \lambda \frac{f(u, D)}{|D|} + (1 - \lambda) \frac{f(u)}{f(.)}, \tag{7.3}$$

where $f(u, D)$ denotes the frequency of term u in D, $|D|$ denotes the length of D, $f(u)$ denotes the collection frequency of u and $f(.)$ denotes the total aggregate of collection frequencies over all terms (collection size). For generating the variants, we set $\lambda = 0.6$ as per the recommendations in previous studies [51]. As a note for practical implementation, the co-occurrence matrix of Equation 7.2 can be efficiently implemented by squaring the sparse term-document matrix of the top retrieved M documents, $X \in \mathbb{R}^{M \times |V_M|}$, i.e., yield the desired $|V_M| \times |V_M|$ matrix with the operation $C = X^T X$.

Weighted Graph of Local Co-occurrences

The co-occurrence matrix constructed from each term pair co-occurrence likelihood of Equation 7.2 represents the adjacency matrix, C, of a graph of $|V_M|$ nodes (each node corresponding to a word). The weight between a pair of nodes in this graph indicates the co-occurrence likelihood between the words (Equation 7.2). A subset of these nodes constitutes the original query terms, i.e. members of the set Q. The rest, i.e. $|V_M - Q|$, is comprised of candidate terms that could be selected for forming the query variants. Formally using the definition of $P(u, v; Q, \mathcal{M})$ from Equation 7.2,

$$G = (Q \cup (V_M - Q), \{(u, v, \omega_{u,v})\}) : \omega_{u,v} = P(u, v; Q, \mathcal{M}) > 0. \tag{7.4}$$

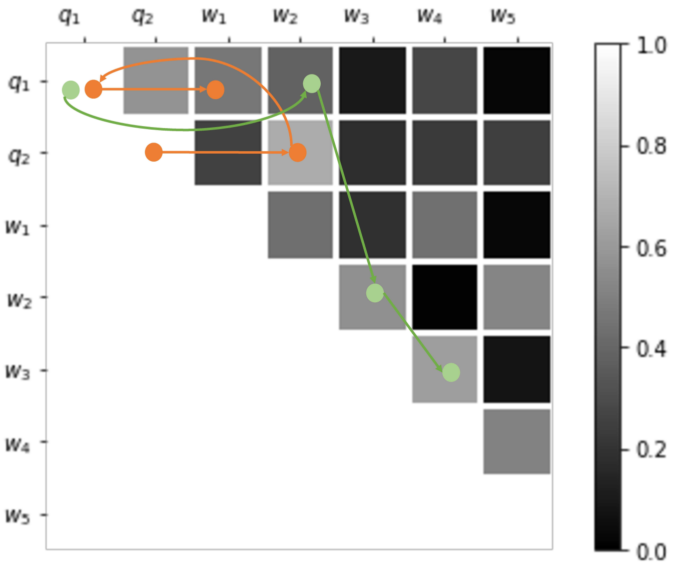


Figure 7.1: A schematic visualization of query variant construction with the help of random walks. Two sample walks of length 4 each are shown in two different colors. The orange colored walk starts from the query term q_2. The walk then visits node (word) w_2 (a word which has a relatively high co-occurrence likelihood with q_2 as seen from a light shade of gray). The walk then continues to q_1 and terminates at w_1 thus generating a variant, $\hat{Q}_1 = \{q_2, q_1, w_2, w_1\}$ of the original query $Q = \{q_1, q_2\}$.

Random Walk for Query Variant Generation

To select a candidate query variant, we initiate a random walk from one of the query nodes chosen with a uniform probability (this ensures that we include at least one query term in the automatically constructed variant). We continue the walk for a small number of steps (specifically, 3-7 in our experiments). Each walk comprises a set of nodes, the corresponding words of which forms a query variant (strictly speaking, a walk is a sequence of nodes; however, in an IR setup, a query is treated as a set rather than as a sequence of terms). We employ a greedy approach to construct the query variants. In particular, the probability of visiting the next node in the walk is Markovian, i.e., it depends only on the current node visited. The probability of selecting the next node (i.e. that of including the next term in a query variant) is given by the maximum likelihood estimate of the neighboring edge weights. This makes it more likely to select a term that has a high co-occurrence likelihood with the most recent term selected. Formally,

$$P(t_i = v | t_{i-1} = u, \ldots, t_1) = \frac{\omega(u, v)}{\sum_{w \in \mathcal{N}(u)} \omega(u, w)}, \quad P(t_1 = q) = \frac{1}{|Q|} \tag{7.5}$$

where $\mathcal{N}(u)$ denotes the neighborhood (adjacent set of nodes) of the current node u, and t_i denotes the i^{th} term added to the walk.

A schematic illustration of the random walk process of query variants generation is shown in Figure 7.1. For the purpose of illustration, the figure shows a sample weighted graph visualized as the part above the diagonal of a local co-occurrence matrix (the part to the bottom-left of the diagonal is left blank to avoid confusion). While one of the walks leads to a query variant that also includes both the original query terms (the orange colored walk $\hat{Q}_1 = \{q_2, q_1, w_2, w_1\}$), the green colored walk ($\hat{Q}_2$) is comprised of only one term from the original query.

Characteristics of the Query Variants

Since during each step of the the walk (Equation 7.5), it is likely to select a word that has a high co-occurrence likelihood with the current word (and by transitivity, also with each word that has already been visited), the set of words eventually included in a walk is likely to represent a query variant that is expected to be semantically related to the original query Q. As the walk proceeds by adding a node at each step to the sequence of nodes already visited, it can happen that a node is visited multiple times. In our query processing stage (Section 7.3.4), the sequence representation of a walk is transformed into a set representation of terms.

The equivalence class of a query generated by this stochastic random walk is likely to constitute a fair mixture of both specializations and generalizations of the original query. It may happen that some query variants contain the original query as a part of them, e.g. the orange colored walk of Figure 7.1. The additional terms in these queries is likely to specialize the information need of the original query [17]. Some queries, on the other hand, contain only a subset of the original query terms and hence is likely to lead to generalizing the information need.

Random Walk Length

While each query variant should seek to address the same information need as that of the original query, it should also contain additional semantically similar terms that could potentially enrich the information need (without drifting it away from the information need of the original query). This requires a careful trade-off between *exploitation* (utilizing what has been constructed till the current stage) and *exploration* (seeking to explore more terms to construct more variants). While too conservative an exploration (a short and compact random walk) may result in a small number of variants to be constructed (thus leading to a small post-feedback effect), a too ambitious exploration (a long and spread walk) may result in a large number of variants, the information need of most of which may in fact be substantially different from that of the original query.

With a manual inspection and some of the initial trends in our experiments, we found that a walk length of 7 works well in practice. Moreover, we set the number of generated query variants (each with a separate instance of a random walk) to 50 after observing a set of initial trends in the feedback results. Since we eventually use each query variant to retrieve ranked lists of documents to aggregate retrieval rank likelihoods of documents, too large a number of variants would contribute to increased run-times, as a result of which the number of variants was set to a modest value of 50.

7.3.4 Query Variants to Feedback Documents

Combining Evidences from Query Variants

After describing the method of automatically constructing query variants, we now describe how to make use of these variants for improving the effectiveness of relevance feedback. The fact that the information need of a manually formulated query variant is quite similar to that of the original query contributes to the effectiveness of feedback and the QPP models [61, 101] that use these variants. However, in the absence of the manually annotated variants (which in fact is representative of a more realistic situation), it is likely that the automatically constructed ones may potentially contain a number of terms that could cause a drift in the information need. This necessitates developing a more robust approach of combining the information retrieved with these query variants. To do so, rather than relying on using a single query variant at a time and then eventually combining their feedback models [61], we instead, for each query Q make use of the *entire* set of its automatically generated variants $\hat{\mathcal{Q}}$, to aggregate a collective belief about the usefulness of a document for relevance feedback.

Retrievability based Document Selection

We now describe how, starting with an equivalence class of automatically generated queries, we obtain a candidate set of documents that could be used for relevance feedback. Specifically, we make use of the concept of *retrievability* [6], which is a quantitative score associated with the likelihood of a document D to be retrieved within the top-M ranks in response to a set of queries sampled from a collection. In the context of our problem, the notion of the collection corresponds to the local set of the top-M retrieved documents. Formally,

$$s(D, \hat{\mathcal{Q}}) = \sum_{\hat{Q} \in \hat{\mathcal{Q}}} r(D, \hat{Q}), \tag{7.6}$$

where $r(D, \hat{Q})$ is the rank at which document D is retrieved for a query variant $\hat{Q}$. For implementation purpose, we retrieved the top-1000 documents for a query, and $r(D, \hat{Q})$ is set to 1001 if D is not retrieved within top-1000.

Intuitively, Equation 7.6 aggregates for each document D, the ranks at which each query variant $\hat{Q}$ retrieves D. A low value of these aggregated ranks (lower the better) for a particular document, say D, indicates that D is retrieved *towards top-ranks for a large number of query variants*. These aggregated rank values are then used to preferentially select documents for relevance feedback with the hypothesis that the documents with small (better) values of aggregated ranks are the ones that are consistently retrieved at top ranks for a large number of query variants. This in turn accumulates evidence for the belief that these documents are strongly related to the information need of the original query and hence should be useful for relevance feedback. While on one hand, consistency in the top-retrieved documents for the good quality query variants may help to select the relevant documents, this way of aggregation is also expected to *discount the noisy contributions* from the (possibly) drifted variants on the other.

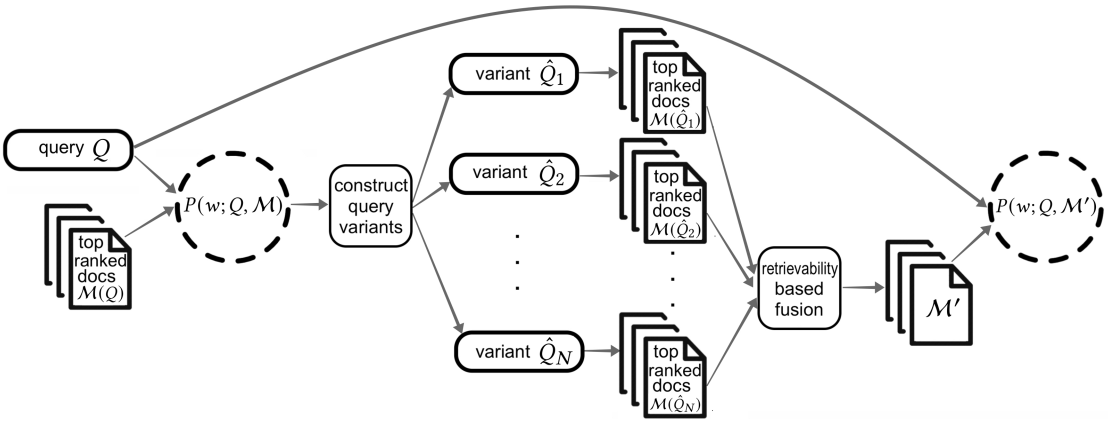

query Q
top ranked docs $\mathcal{M}(Q)$
$P(w;Q,\mathcal{M})$
construct query variants
variant $\hat{Q}_1$
variant $\hat{Q}_2$
variant $\hat{Q}_N$
top ranked docs $\mathcal{M}(\hat{Q}_1)$
top ranked docs $\mathcal{M}(\hat{Q}_2)$
top ranked docs $\mathcal{M}(\hat{Q}_N)$
retrievability based fusion
$\mathcal{M}'$
$P(w;Q,\mathcal{M}')$

Figure 7.2: A schematic workflow diagram of our proposed weakly supervised relevance model (WSRM).

Differences with the existing notion of ***retrievability***

The notion of retrievability that we use in Equation 7.6 is different in two ways from its original definition [6]. First, in [6] it relied on a parameter r_{max} that specified the upper bound of the rank, and second, it accumulated Boolean values (1/0) indicating whether the rank of a document was within this specified bound. In our approach, firstly, we do not restrict the rank computation with a bound (because for PRF it is difficult to foresee the rank cut-off). Secondly, we aggregate the rank values themselves instead of the Boolean indicator variables to get a better estimate of the likelihood, which also makes the overly restrictive rank cut-off unnecessary.

Feedback with Selected Documents

Next, we sort the feedback (top-M) documents in ascending order of the aggregated retrievability (rank aggregated) scores computed by Equation 7.6. The top-M' documents from this set are then used for relevance feedback, where M' is a parameter. PRF with this set of documents thus combines evidences across a range of different queries and is thus expected to yield better retrieval effectiveness. In particular, we employ RM (Equation 7.1) on the top-M' documents from this set. The parameter M' is independent of M, the number of documents used to compute the local co-occurrence graph (Equation 7.2) and the random walk on it (Equation 7.5).

We call our model the 'Weakly Supervised RM' (**WSRM**) because the retrieval position likelihoods captured with the aggregated retrievability scores (Equation 7.6) act as weak supervision signals for estimating document relevance. A schematic overview of the relevance feedback workflow for WSRM is depicted in Figure 7.2.

7.3.5 Manually Annotated Query Variants

The main advantage of our proposed feedback method is that it *does not need to rely on manually formulated query variants*. Since recent studies have shown that manually annotated query variants are useful to improve the effectiveness of relevance feedback and query performance prediction (QPP) [14, 61], we in our proposed feedback method (WSRM), also incorporate information from manually formulated query variants. For this, instead of estimating the co-occurrence weights on the top-M retrieved documents (Equation 7.2), we adapt the idea of [61] where separate relevance models are estimated with each manually constructed individual query variant. Consequently, instead of a single set of top-retrieved set of documents, $\mathcal{M}$, we obtain a total of N such different sets of documents one for each query, where N denotes the number of manual query variants. We then compute the local co-occurrence weights by aggregating the evidences from the top-M documents of each query, i.e.,

$$P(u, v; Q_1, \ldots, Q_N, \mathcal{M}_1 \ldots, \mathcal{M}_N) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{M}_i|} \sum_{D \in \mathcal{M}_i} P(u|D)P(v|D). \tag{7.7}$$

Similar to the single query input, these local co-occurrence values of Equation 7.7 are used to define a graph with weighted edges, the only difference being that the random walk can now start from an arbitrary query term in any of the manual variants. The rest of the methodology is the same, i.e., we use the retrievability based rank aggregation mechanism (Equation 7.6) to construct the final set of feedback documents, $\mathcal{M}'$.

7.4 Weak Supervision with Priors

In this section, we describe how the weakly supervised relevance model proposed in Section 7.3 can be applied in the case of POI (point-of-interest) recommendation, where additional prior beliefs about the contextual appropriateness of a term can act as weak signals to improve RM estimation. POI recommendation, being a precision-oriented task [1], provides an interesting use-case to study the robustness effects of relevance feedback.

7.4.1 Contextual Recommendation

In an IR-based contextual POI recommendation framework, a system needs to return a ranked list of POIs based on a user's preference history and also his current contextual constraints. Examples of contextual constraints include the current location of the user, the purpose of the trip such as 'holiday' etc. To draw an analogy from the problem of contextual POI recommendation to that of IR, it can be considered that the user preference history and the current contextual constraints in a recommendation system are analogous to the notion of a query in IR, whereas the candidate POIs are analogous to documents [23, 20] (Chapter 3 - 6).

Here we first briefly revise our previous work from Chapter 3 - 6, and then explain the novel contribution of this chapter. Specifically, a query in contextual recommendation problem is personalized in nature, comprising a) a description of the POI that the user has visited in the past, b) the reviews posted by the user on location-based social networks and the tags associated with the reviews, and c) the ratings associated with the past POI visits. In addition, each query is also associated with a current location of the user, which imposes a *hard constraint* that the recommended POIs must be from the current location of the user. Furthermore, a query also contains a list of *soft constraints*, corresponding to a list of categorical values representing *trip qualifiers*, e.g., 'trip-type = {business, holiday,...}', 'trip-duration = {day-trip, night-out,...}', 'accompanied-by = {alone, family,...}' etc.

Following the work of [23, 20] (Chapter 3 - 6) we represent a query as a structured document of the form (t_u, q_u) comprised of the review-text or tags from the user profile and the trip-qualifier contexts, respectively. A two-step factored RM-based approach that uses both the query and the top-retrieved documents was proposed in [23, 20] to obtain a combined RM of the form

$$\begin{aligned} P(w|\theta_{q_u}) &= \sum_{d \in D_u} rP(w|d)\psi(w, q_u) \prod_{t \in t_u} P(t|d) \\ P(w|\theta_{q_u,l_u}) &= \sum_{d \in D_M(\theta_{q_u}):L(d)=l_u} P(w|d)\psi(w, q_u) \prod_{t \in \theta_{q_u}} P(t|d), \end{aligned} \tag{7.8}$$

where D_u denotes the set of documents (POIs that the user had visited in the past), $L(d) = l_u$ lists the candidate set of POIs in the current location (the hard constraint), $D_M(\theta_{q_u})$ denotes the top-M retrieved POIs with θ_{q_u} as the expanded query, $P(w|d)$ denotes the normalized term frequency of a word w in document d, and $\psi(w, q_u)$ denotes a prior belief on a *contextual appropriateness score* of a term w with respect to a context term q_u (which is explained later in Equation 6.1).

To see how our proposed relevance feedback framework may be useful to estimate $P(w|\theta_{q_u})$ (Equation 7.8), note that the first step of constructing the local co-occurrences graph (Section 7.3.3) can be substituted with that of leveraging information from the co-occurrence graph of the prior beliefs of manually annotated contextual appropriateness scores between term pairs ($\psi(w, q_u)$ of Equation 7.8). Next, we describe how to generate the query variants with random walk applied on the graph of binary relations of the term appropriateness scores.

7.4.2 Query Variants with a Knowledge-base

Knowledge-base to Weighted Graph

A knowledge resource of term-category associations was compiled in [3], which comprises lists of pairs constituting a term and a non-location trip-qualifier with manually judged relevance scores of the form (t, q, a), where t is a term (e.g. `food`), q is a single category (e.g. `holiday`) and $a = 1(a \in [0, 1])$ is the appropriateness score. An example of a non-relevant pair with a lower score is `(nightlife, business, 0.1)`. We formally denote this knowledge resource as

$$\kappa : (w, q) \mapsto [0, 1], w \in V, q \in Q_i, i \in \{1, \ldots, c\}), \tag{7.9}$$

where Q denotes the set of *joint* non-location type contexts, Q_i denotes a context category, and V denotes the vocabulary set of the review text and tags. For a given non-location contextual constraint vector q_u in the user query, we use embedded word vector representations to aggregate the similarities of each word in the review text/tag of a user profile with the seed words assessed as relevant for a context q_u. Formally, $\forall w \in P_U$ we define a function,

$$\psi(w, q_u) = \max(\mathbf{w} \cdot \mathbf{s}),\ s \in \cup\{t : \kappa(t, q_U) = 1\}. \tag{7.10}$$

Equation 6.2 indicates that for each word w (embedded vector of which is represented as $\mathbf{w}$) contained in the text from the profile of a user, we compute its maximum similarity over a subset of seed words relevant only for the given context, i.e., the words for which $\kappa(s, q_U) = 1$. In our experiments, we make use of the `word2vec` (skipgram algorithm) [68] for the purpose of embedding the vector representation of a word.

The reason for using the maximum as the aggregate function in Equation 6.2 is that a word is usually semantically similar to a small number of seed set of words relevant for a given context. To illustrate this with an example, let the 3-dimensional query context comprising trip-type, duration and company be set to the value of ‘(vacation, day-trip, friends)’. The relevant seed set in this example constitutes words such as ‘base-ball stadium’, ‘beer-garden’, ‘salon’, ‘sporting-goods-shop’ etc. However, a word such as ‘pub’ is similar to only one member of this seed set, namely ‘beer-garden’, which means that other aggregation functions, such as averaging, can lead to a low aggregated value, which in this case is not desirable.

WSRM with Edge Weights from Knowledge-base

The values indicating term pair relations, $\psi(w, q_u)$, computed by Equation 6.2 are then used to define a weighted graph (similar to the one of Equation 7.4). After defining the graph this way, we then apply the random walk based method (Equation 7.5) to initiate a number of different walks from the query terms. In this case, therefore, the walks are comprised of tags and trip qualifier terms.

As a novel contribution of this chapter different to that of [23, 20], we then modify the RM estimation of Equation 7.8 with the weak-supervised approach based on query variants. Specifically, instead of applying RM over the top-M retrieved documents $D_M(\theta_{q_u})$, we use the documents with the lowest rank aggregation scores obtained from the query variants (Equation 7.6). This weak supervised RM is able to take into account the prior beliefs in the contextual appropriateness of terms from a knowledge resource.

7.5 Experimental Setup

We evaluate our PRF approach on two different tasks - a) standard ad-hoc IR, where our proposed feedback algorithm works with the automatic query variants generated with the local co-occurrence information (WSRM), and b) POI recommendation, where we leverage information from term-level contextual appropriateness scores to formulate the query variants (WSRM-KB).

7.5.1 Dataset

For the ad-hoc task, we performed our experiments on TREC 6-8 and Robust topic sets comprising 150 and 99 topics respectively. The target documents collection is TREC ad-hoc IR collection from disks 4 and 5 without the congressional records. A summary of the dataset is shown in Table 2.1. For the POI recommendation task, we use the TREC-CS 2016 dataset (phase-1 setup) [43]. The task requires a system to return a ranked list of 50 POIs from a given query collection (user profiles), that best fit the user preference history and the user's current contextual constraints. A user's contextual constraint is a 3-dimensional vector of categorical values (corresponding to non-location type trip qualifiers) as outlined in Table 3.2. The overall collection comprises over 1.2M of POIs in total, and the number of context queries used in our experiments is 61 (part of the TREC-CS 2016 dataset).

UQV Dataset for manually obtained query variants

In a more realistic use-case, the only information available to an IR model is a single query (as entered by a user). Our PRF algorithm WSRM constructs the variants automatically by employing random walks on the local co-occurrence matrix of top retrieved documents. Recent literature has investigated the effectiveness of feedback models on manually formulated query variants, e.g. using the UQV dataset. In this dataset, given a manually constructed back-story (a narrative illustrating the information seeking situation) corresponding to a TREC query, participants were asked to formulate queries. These queries were then post-processed (e.g. duplicates removed, spelling errors corrected etc.) and released as a resource for the purpose of conducting experiments with query variants.

Although the pre-existence of query variants represents a somewhat unrealistic experiment setup, nonetheless for the sake of comparing our proposed feedback approaches with the other feedback methods reported in the literature, e.g. [14, 61], we also conduct PRF experiments on manually formulated variants.

7.5.2 Baselines and Parameter Settings

Single-Query Baselines

Some of the standard baseline approaches are only able to make use of a single query for retrieving a ranked list of documents. These baselines include **BM25** and the standard relevance model, **RM** ('RM3' version) [51, 48].

Top-Document Set Permutation Baselines

Instead of blindly assuming that the top-M retrieved documents are useful for relevance feedback, our method essentially relies on permuting this set of top documents (based on the rank aggregation scores of Equation 7.6) and select a new top set (M') of documents for feedback. To demonstrate the effectiveness of this method, we undertake a number of baselines that employ some form of a document reordering mechanism to choose a set of documents, different from the top-retrieved ones.

A simple such permutation function is to sort the top-M retrieved set of documents by document length, and then select the top M' ones for feedback ($M' < M$). Since the input to the selection function is the set of documents that are retrieved within the top M ranks, they have high similarity scores with the query. A further filtering based on their lengths may serve as a useful heuristic to choose the ones that could potentially improve feedback. A different choice of the permutation order yields two different baselines.

1. 'Shortest Document First' (**SDF**), which assumes that the shortest documents will be more useful for feedback because they are more likely to be focused on the query topic.
2. 'Longest Document First' (**LDF**), which assumes that the longest documents will be more useful for feedback because they are likely to contain a higher number of terms that eventually could be useful to enrich the initial query.

Clustering-based Resampling Baseline

The clustering based resampling method, proposed in [52, 9], employs a document neighborhood induced permutation on the top retrieved M documents. Specifically, the method involves finding neighborhoods of documents (called 'overlapping clusters' by the authors of [52]). The method assumes that *dominant* documents for a query are the ones with several nearest neighbors with high similarities, i.e., the neighborhoods with the highest aggregated retrieval scores (essentially assuming that such a neighborhood effectively represents the core topic of the information need). Since this cluster based resampling method estimates a new set of documents that is used for feedback, we employ this approach as another baseline, which we call '**kNN**'.

In fact, in addition to selecting the documents for feedback, since the cluster-based resampling method also involves making use of a cluster-based smoothed query likelihood model, for a fair comparison with our approach, we incorporate the neighborhood-based smoothed mechanism for computing the maximum likelihood estimates (MLEs) of the local co-occurrences (Equation 7.2). Specifically, instead of using Equation 7.3 for computing the MLEs at the level of documents, we employ

$$P(w|\mathcal{C}) = \lambda \frac{f(w,\mathcal{C})}{|\mathcal{C}|} + (1-\lambda)\frac{f(u)}{f(.)}, \tag{7.11}$$

which differs from Equation 7.3 in that it samples terms from the bag-of-words representation of a neighborhood (overlapping cluster), $\mathcal{C}$, of documents. Applying Equation 7.11 for computing the local co-occurrence graph, subsequently followed by a random walk based query variant generation and rank aggregation for selecting the feedback document set, constitutes *a variant of our proposed method* for relevance feedback, which we call 'Cluster-based Weakly Supervised RM' (**CWSRM**).

Fusion Baselines for Single and Multi-Queries

A recent work [61] shows that both the approaches of - a) combining separate relevance models estimated with each input query (variant) as a single feedback model **AriRM**, and b) separately executing feedback

	Params	Development Set		
	tuned on	TREC 8		
Method	dev set	P@5	P@10	MAP
BM25	k=0.5, b=0.5	0.4960	0.4780	0.2619
RM	M=3, T=160	0.5360	0.5020	0.2803
SDF	M=20, T=160	0.5200	0.4960	0.2685
LDF	M=20, T=160	0.4400	0.4000	0.2429
kNN	$\|\mathcal{C}\|$=2, T=100	0.5680	0.5200	0.2847
AriRM	M=3, T=160	0.5360	0.4760	0.2480
MultiRM	M=3, T=160	0.5280	0.4820	0.2392
WSRM	M'=3, T=160	0.5680†‡	0.5200‡	**0.2887**
CWSRM	M'=3, T=160	**0.6000***†‡	**0.5340**†‡	0.2849

Table 7.1: Ad-hoc IR relevance feedback experiments with *single queries as input*, i.e. without using manually annotated query variants on the TREC ad-hoc IR (TREC 8) topic sets. Parameters for each method were tuned separately on TREC 8, and then each method was tested on the remaining topic sets with the optimal parameter configurations. Statistical significance of the proposed methods (WSRM and CWSRM) in comparison to the three most effective baselines - kNN, RM and AriRM, are denoted with '$*$', '$\dagger$' and '$\ddagger$', respectively (t-test with $p = 0.05$).

models on the individual query variants and then finally merging the results **MultiRM**, improve retrieval effectiveness. To investigate if our proposed rank aggregation method of document selection for relevance feedback is effective, as baselines we employ the fusion based approaches AriRM and MultiRM for both single query setup and multi-query setup (i.e. with and without the UQV query variants for the TREC topic sets). In the single query setup ($N = 1$), we applied AriRM and MultiRM on query variants that were generated automatically by our proposed approach. For the multi-query case, we made use of only the supplied query variants from the UQV dataset alone (inclusive of the original TREC query) to fuse the feedback model (AriRM), or the result-lists (Multi-RM).

The parameters of each method, namely - a) (k, b) for BM25, b) number of clusters, $|\mathcal{C}|$ for kNN, c) the number of feedback documents and terms, (M, T) respectively, for RM, kNN, AriRM, and MultiRM, and d) the number of feedback documents (in the second-stage after document selection) and terms (M', T) respectively for WSRM and CWSRM - were tuned individually by grid search on the TREC-8 dataset with respect to the metric P@5. The decision to use TREC-8 as the development dataset was arbitrary. The optimal parameter settings (as obtained on the development dataset) were then applied for each method on the rest of the topic sets, namely TREC 6, 7 and Robust.

	Params	Test Set								
	tuned on	TREC 6			TREC 7			TREC Rb		
Method	dev set	P@5	P@10	MAP	P@5	P@10	MAP	P@5	P@10	MAP
BM25	k=0.5, b=0.5	0.4680	0.4280	0.2306	0.4760	0.4400	0.1943	0.5051	0.4404	0.2896
RM	M=3, T=160	0.4680	0.4400	0.2504	0.5080	0.4840	0.2284	0.5354	0.4657	0.3292
SDF	M=20, T=160	0.4520	0.4220	0.2343	0.4200	0.4000	0.1928	0.4909	0.4465	0.3004
LDF	M=20, T=160	0.3680	0.3240	0.2019	0.4240	0.3720	0.1834	0.4061	0.3455	0.2332
kNN	$\|\mathcal{C}\|$=2, T=100	0.4600	0.4320	0.2455	0.4840	0.4340	0.2186	**0.5576**	**0.4859**	0.3377
AriRM	M=3, T=160	0.4600	0.3880	0.2170	0.4640	0.4380	0.2222	0.5212	0.4465	0.3076
MultiRM	M=3, T=160	0.4440	0.3960	0.2164	0.4680	0.4220	0.2142	0.5111	0.4303	0.2996
WSRM	M'=3, T=160	**0.5120***†‡	**0.4540***‡	**0.2600**	**0.5320***†‡	**0.4880***‡	**0.2387**	**0.5576**†‡	0.4727	0.3340
CWSRM	M'=3, T=160	0.4840*‡	0.4360‡	0.2480	0.4840	0.4540	0.2205	0.5455	0.4808	**0.3415**

Table 7.2: Ad-hoc IR relevance feedback experiments with *single queries as input*, i.e. without using manually annotated query variants on the TREC ad-hoc IR (TREC 6, TREC 7, and TREC Rb) topic sets. Parameters for each method were tuned separately on TREC 8, and then each method was tested on the remaining topic sets with the optimal parameter configurations. Statistical significance of the proposed methods (WSRM and CWSRM) in comparison to the three most effective baselines - kNN, RM and AriRM, are denoted with '*', '†' and '‡', respectively (t-test with $p = 0.05$).

7.5.3 POI Recommendation Settings

Similar to the ad-hoc IR setup, for contextual suggestion we also employ BM25 and RM as the standard baselines. Since a factored version of relevance model (FRM) has been shown to be effective for the contextual suggestion task [23, 20], we employ this method as one of our baselines. Concretely speaking, FRM [23, 20] first enriches the user history and tags to better match the POI descriptors, and then follows this up with a standard RM feedback on POI descriptors using this enriched user history.

To investigate if rank aggregation on automatically generated query variants can improve FRM, we investigate two variants of the weak supervised RM (WSRM) for the contextual suggestion experiments. First, we investigate **WSRM**, which uses Equation 7.10 to constitute the query variants by leveraging information from the knowledge base of manually assessed appropriateness scores.

As the second variant of our proposed approach for contextual recommendation, we investigate **WSFRM**, which is the factored counterpart of WSRM (as FRM is to RM), i.e. instead of applying WSRM only for generating query variants and rank aggregating the retrieved POIs for better feedback document (POI description) selection, we also apply WSRM to enrich the information in the user context as well. The weights in the local co-occurrence graph are estimated with the ψ function representing the manually annotated contextual appropriateness scores (Equation 7.10). The parameters for each method were tuned

Query:	foreign minor germany
Variants:	foreign germani feder european minor dai govern germani romania mar great minor practic poland minor polici type union econom past kinkel
Query:	behavioral genetics
Variants:	behavior determin problem thoma state time genet twin genet gene embryo part behavior time children behavior profil parent genet famili environ

Table 7.3: Examples of automatically constructed query variants (stemmed words) for two sample queries of TREC 8.

independently by conducting a grid search with respect to the metric nDCG@5.

7.6 Results

In this section, we present our experimental results of both the ad-hoc retrieval task, and the task of contextual POI recommendation.

7.6.1 Ad-hoc IR Experiments

Without Manual Query Variants

From Table 7.1 and 7.2, we observe the following. First, although RM improves MAP substantially for each topic set, it is seen that the improvements in $P@5$ are marginal even when compared to a relatively simple baseline such as SDF (e.g. compare the TREC-8 $P@5$ values for RM and SDF). This indicates that a more effective approach may potentially improve precision at top ranks even further. This, conforming to observations of previous studies [52, 91], also confirms that a more robust document selection approach could potentially improve PRF quality.

Second, it is observed that the baseline method kNN [52, 9] is able to substantially improve precision at top ranks (as compared to RM). This reinforces the importance of effectively selecting the set of feedback documents. In fact, our proposed method, WSRM, achieves comparable results with that of kNN. However, an important point to observe is that kNN does not generalize well to the test topic sets (TREC 6 and 7), which indicates that this method is overly sensitive to the choice of its parameters. On the other hand, the facts that WSRM achieves similar effectiveness on the development set and that it also generalizes well on the test data indicates that WSRM is more resilient to parameter variation effects. This also confirms that leveraging information from *rank aggregation statistics* offers a better way to select the candidate set of documents for relevance feedback in comparison to the *neighborhood-based*

Dataset	Method	Parameters	P@5	P@10	MAP
TREC 6	AriRM	$M = 3, T = 160$	0.5840	0.5160	**0.2882**
	MultiRM	$M = 3, T = 160$	0.5760	0.4920	0.2823
	WSRM	$M' = 3, T = 60$	**0.6000**†	**0.5220**†	0.2757
TREC 7	AriRM	$M = 3, T = 160$	**0.6680**	**0.5760**	**0.3000**
	MultiRM	$M = 3, T = 160$	0.6640	0.5660	0.2939
	WSRM	$M' = 3, T = 60$	0.6400	0.5620	0.2666
TREC 8	AriRM	$M = 3, T = 160$	0.6360	0.5800	**0.3279**
	MultiRM	$M = 3, T = 160$	0.6280	0.5840	0.3233
	WSRM	$M' = 3, T = 60$	**0.6600***†	**0.5920**	0.3170
TREC Rb	AriRM	$M = 3, T = 160$	**0.6828**	**0.5848**	**0.4237**
	MultiRM	$M = 3, T = 160$	0.6707	0.5727	0.4110
	WSRM	$M' = 3, T = 60$	0.6586	0.5556	0.3901

Table 7.4: Comparisons between WSRM and AriRM/MultiRM on the UQV manual query variants. Significance of WSRM (t-test with $p = 0.05$) is shown with * (AriRM) and † (MultiRM).

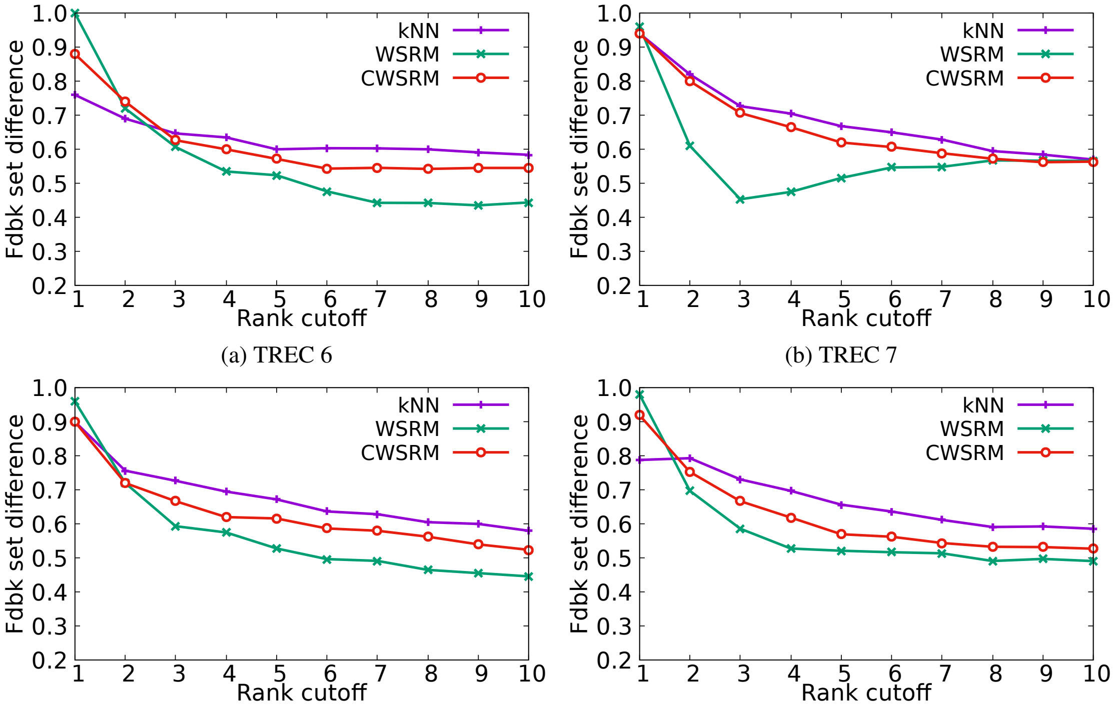
estimation in kNN for a document's likely usefulness for feedback.

Third, somewhat to our surprise, we observed that the retrieval effectiveness of the pre-fusion and post-fusion based feedback methods (i.e., AriRM and MultiRM respectively) was not satisfactory (compare the AriRM and MultiRM results with those of RM for each topic set). This corroborates the fact that fusion based approaches tend to work well with manually annotated query variants, when each query variant points to the exact same information need.

Finally, we observe that the neighborhood based smoothing of [52] for estimating the local co-occurrences eventually help to further improve the quality of relevance feedback (as can be seen from the CWSRM results of Table 7.1 and 7.2 in comparison to the WSRM ones). However, the combination method does not generalize well for the test sets of topics. This happens due to the percolating parameter sensitivity effect of kNN method onto CWSRM. As an illustrative example for the quality of the automatically generated query variants, Table 7.3 shows these variants obtained with WSRM on two TREC-8 topics.

With Manual Query Variants (UQV data for TREC Robust)

Table 7.4 shows that with a small number of query variants, the fusion-based approaches usually work well in practice. We failed to notice any consistent trends in the results from Table 7.4. A reason for this could be the fact that since manual variants are *good quality* alternate representations of an information need, the results achieved by the fusion based models exhibit a *saturation effect* in the results making it



(a) TREC 6

(b) TREC 7

(c) TREC 8

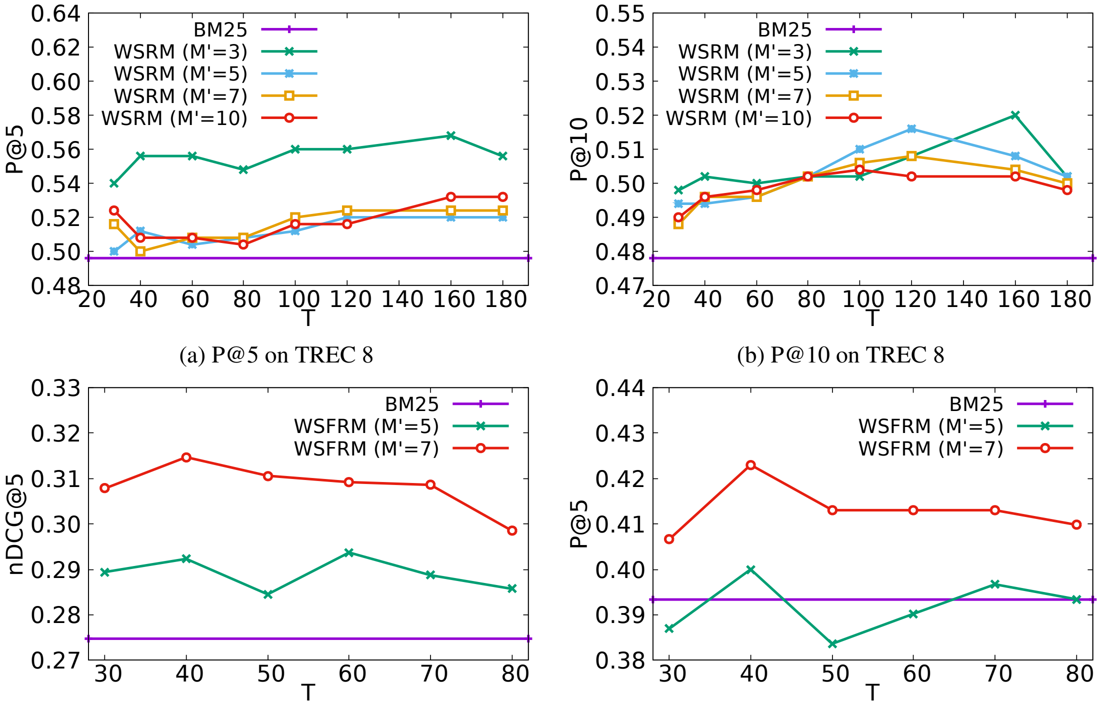
(d) TREC Rb

Figure 7.3: Comparison of set differences in top ranked documents for 3 different feedback document scoring methods - kNN, WSRM and CWSRM, at specific rank cutoffs, i=1,. . .,10, shown on the x-axis. The set difference values (y-axis) are computed as $(\mathcal{M}'_i - \mathcal{M}_i)/i$, where $\mathcal{M}_i$ ($\mathcal{M}'_i$) represents the set of top-i documents before (after) document re-scoring.

difficult to further improve them with automated processing. Despite this saturation effect, some of the results show improvements in a couple of cases, e.g. we notice that WSRM leads to an improvement in the precision at top ranks on TREC-6 and TREC-8 topic sets. As a point of note, it is worth noting that the experiments reported in Table 7.4 represent a rather unrealistic situation for ad-hoc IR, because it unlikely for a user to enter a number of synonymous representations of his information need.

Feedback-Document Set Analysis

Existing literature has shown that it is necessarily true that either a well filtered set of top-M documents or the set of true relevant documents are the most effective to improve retrieval effectiveness [91, 52, 9]. An interesting question then is to investigate how many new (yet effective) documents, on an average, is a document selection strategy able to bring within the top M' ranks which eventually leads to the improvements in retrieval effectiveness as demonstrated by the WSRM results in Table 7.1, 7.2. In other words, as per our terminology, the question becomes - what is the difference between the sets $\mathcal{M}$ and $\mathcal{M}'$? A high value of this difference indicates that a feedback document selection algorithm is able to leverage information even from outside the initial set of top-M documents thus attributing the reasons for



(a) P@5 on TREC 8

(b) P@10 on TREC 8

(c) nDCG@5 on TREC-CS

(d) P@5 on TREC-CS

Figure 7.4: Effect of precision at top ranks (P@5, P@10 for TRCE 8, and nDCG@5, P@5 for TREC-CS) with respect to changes in #feedback documents (M') and #expansion terms (T) used in WSRM (for TREC 8) / WSFRM (for TREC-CS) estimation.

improvements to this difference.

Figure 7.3 shows the differences between the two sets $\mathcal{M}$ (top-M of initial retrieval, in our case, BM25) and $\mathcal{M}'$ (top-M' after document re-scoring) as obtained by the three feedback document selection methods, namely kNN, WSRM and CWSRM. These differences are measured at a number of different rank cut-off points. The results show that both kNN and (C)WSRM are able to retrieve a fair number of new feedback documents (outside the initial top-M). However, the better MAP values of (C)WSRM (Table 7.1, 7.2) indicates that both WSRM and CWSRM achieve the desired trade-off between exploration (leveraging information from new documents) and exploitation (making use of the top-M set). The method, kNN, on the other hand, leads to a more aggressive exploration, which eventually yields lower $P@5$ and MAP values (as seen from Table 7.1, 7.2).

Sensitivity Analysis

Figure 7.4a and 7.4b show the parameter sensitivity effects of WSRM on precision at top ranks for the development set, i.e. TREC-8 topics. We observe that selecting a small number of feedback documents after re-scoring helps achieve the best results, which in turn shows that our approach of document selection

Method	Optimal Params.	nDCG@5	nDCG	P@5	MAP
BM25	$k = 1.1, b = 0.3$	0.2747	0.2889	0.3934	0.1326
RM	$M = 5, T = 25$	0.2615	0.3091	0.3574	0.1437
FRM	$M = 5, T = 25$	0.2919	0.3418	0.3934	0.1616
WSRM	$M' = 5, T = 30$	0.2746	0.3214	0.3738	0.1520
WSFRM	$M' = 7, T = 40$	**0.3147***	**0.3576***	**0.4230***	**0.1727***

Table 7.5: Comparisons between our proposed approaches (WSRM and WSFRM) and the baselines on the TREC-CS data. Significance between the differences between WSFRM and FRM is denoted by '$*$' (t-test with $p = 0.05$)

by rank aggregation over query variants is effectively able to filter out useful information for relevance feedback at the very top ranks.

7.6.2 Contextual Recommendation Experiments

Similar to the observations for the ad-hoc task, Table 7.5 shows that our approach improves the POI retrieval effectiveness (particularly, precision at top-ranks) for the contextual recommendation task. It can be seen that our proposed approach, WSRM, and its factor-based variant, WSFRM, outperform both RM and FRM. This indicates that automatic generation of query variants and then using the retrievability measure on them to construct the feedback set works better in the presence of true prior beliefs about term-level relevance. Being a precision oriented task (because real-life use-case requires that results are to be displayed on mobile devices with limited UI resources), it is particularly interesting to observe the improvement of precision for POI recommendation at the top-ranks ($nDCG@5$) with WSFRM. Figure 7.4c and 7.4d show that the trends for parameter sensitivity effects are similar to that of Figure 7.4a and 7.4b.

7.7 Summary

In this chapter, we seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents are useful for relevance feedback. Primary motivation of this chapter is to explore a way of improving precision at top ranks for IR methods (specifically pseudo-relevance feedback methods) in general, which is eventually applied in the specific task of contextual recommendation.

We propose a concept of weakly supervised relevance models by using the notion of *retrievability* from automatically constructed query variants to improve the quality of relevance feedback. We observe that our approach consistently improves precision at top ranks in two different tasks, namely TREC ad-hoc and the contextual POI recommendation.

In particular, results of our experiments show that our proposed method of a weakly supervised relevance model, WSRM (and WSFRM, which is a counter part of our factored model, FRLM) is able to achieve substantially better precision at top-ranks (e.g. higher nDCG@5 and P@5 values) for ad-hoc IR and points-of-interest (POI) recommendation tasks.

For TREC adhoc, although traditional relevance model (RM) improves MAP substantially for each topic set, it is seen that the improvements in 'P@5' are marginal. It is also observed that the kNN-based resampling method is able to substantially improve precision at top ranks (as compared to RM). However, our approach generalizes well on the test data indicates that WSRM is more resilient to parameter variation effects. Somewhat to our surprise, we observe that fusion based approaches tend to work well with manually annotated query variants, when each query variant points to the exact same information need.

Chapter 8

Conclusions and Future Work

In this thesis, we focus on the problem of suggesting ‘points of interest’ (POIs) to a user given her current context(s), by leveraging relevant information from her past preferences. We argue that contextual POI recommendation is essentially a personalized IR task, where personalized content matching is important. In fact, after experimenting with different content-based, collaborative filtering based and hybrid approaches, Arampatzis and Kalamatianos [5] found that the content-based approaches performed better than other approaches for this problem. Following this argument, our proposed approach in this thesis is an IR based content matching one. We hypothesize that it is more suitable to formulate the POI recommendation problem as a *constrained IR* problem, which is characteristically different from the scope of a traditional RecSys approach where the popularity of an item depends only on user ratings (e.g. neural collaborative filtering for movie recommendation [45]), or other contextual features.

Primary objective of this thesis, broadly speaking, is to *explore Information Retrieval (IR) based approaches for contextual POI recommendation, with a particular focus to improve precision at top ranks*. In particular, this thesis proposes a generic relevance feedback based framework for contextual POI recommendation. We gradually build up the overall framework of our proposed model, in increasing order of complexity, by incorporating the following three aspects:

i) *factored relevance modeling* to achieve an optimal combination of the user’s *preference history in past contexts (exploitation)*, and the *relevance of top-retrieved POIs in the user’s current context (exploration)*,

ii) *word semantics* in the form of *kernel density* estimates computed by distances between embedded word vectors of the user tags and the POI descriptors, and

iii) *soft (trip-qualifier) constraints* modeled by leveraging information from a knowledge-base of manually assessed contextual appropriateness of words under the pretext of a given context category, either in separate or in joint forms.

Our experiments on the TREC-CS 2016 [43] dataset show that even the simplest of our proposed class of models (i.e. the factored relevance model) outperforms a range of different baseline approaches involving standard IR or recommender system methodologies. Moreover, it is shown that the additional generalizations in our proposed framework, i.e. including word semantics and information from a knowledge base, further improves POI effectiveness.

In addition, we make an effort to achieve better precision at top-ranks by improving the quality of relevance feedback for IR in general, which is eventually applied in the specific task of contextual POI recommendation. POI recommendation, being a precision-oriented task [1], provides an interesting use-case to study the robustness effects of relevance feedback.

8.1 Revisiting Research Questions, Contributions and Achievements

In this section, we revisit the four research questions, mentioned in Chapter 1, and discuss how they have been addressed through different chapters of this thesis. We also enlist chapter wise contributions, findings and achievements.

8.1.1 Factored ReLevance Model (FRLM)

The first and one of the most significant contributions of this thesis is a novel *Factored ReLevance Model* (FRLM) for contextual POI recommendation. Chapter 3 introduces a standard IR-based research framework where different experiments have been conducted. In Chapter 4, we focus on suggesting 'points of interest' (POIs) to a user given her current location (*hard* contextual constraint), by leveraging relevant information from her past preferences.

The first research question, mentioned in Chapter 1, is reproduced here.

> **RQ 1:** What is an effective and systematic approach to find the trade-off between a user's preference history (*exploitation*) and the information about the POIs constrained to a *hard* contextual constraint such as 'location' (*exploration*) for contextual POI recommendation?

Primary objective of RQ 1 is to investigate a systematic way to make a balance between *exploitation* and *exploration*, given a *hard* location constraint.

As mentioned earlier in this thesis (Chapter 1, 2), a number of studies have investigated the problem of contextual recommendation from the point of view of matching the content between the POI (document) representation and the user profile (query) representation [94, 49]. Note that contextual recommendation systems based on this thread of work mainly rely on *exploiting* the existing preferential knowledge of users from their profiles. On the other hand, a different thread of work [26, 39] utilizes rating-based collaborative filtering, i.e. information from other users to estimate the popularity of a POI in a local context with the hypothesis that POIs with frequent positive ratings from other users could also be relevant to the current user. In contrast to *exploitation*, this collaborative filtering based thread of work primarily relies on *exploring* the POIs using the current context. However, there is no systematic investigation on the use of user's preference history, top rated POIs in the current context or both, while predicting the appropriateness of a POI for a user in her current context.

We hypothesize that, an automated contextual recommendation algorithm is likely to work well if it can extract information from the preference history of a user (*exploitation*) and effectively combine it with information from the user's current context (*exploration*) to predict an item's (POI's) 'usefulness' (relevance) in the new (location) context. To balance this trade-off between *exploitation* and *exploration*, we propose a generic unsupervised framework involving a factored relevance model (FRLM), comprising two distinct components, one corresponding to the historical information from past contexts, and the other pertaining to the information from the local context.

A characteristic of our model is that it achieves a sweet-spot between the user's preference history in past contexts (*exploitation*), and the relevance of top-retrieved POIs in the user's current context (*exploration*). Our experiments on the TREC-CS 2016 dataset show that our proposed model of a factored relevance model is able to effectively combine these two sources of information, leading to significant improvements in contextual recommendation quality.

RQ 1 has been successfully answered, and we conclude that the systematic infusion of *exploitation* and *exploration* (factored relevance modeling) improves the effectiveness of POI retrieval. This part of work is published in ACM SIGIR ICTIR 2019 conference [23].

> Anirban Chakraborty, Debasis Ganguly, Annalina Caputo, and Séamus Lawless. A Factored Relevance Model for Contextual Point-of-Interest Recommendation. 2019. In *Proceedings of The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19)*, Santa Clara, CA, USA. Pages 157 - 164, ACM, New York, NY, USA.

Chapter 4 of this thesis is based on this paper.

8.1.2 FRLM with Word Semantics

The second research question, RQ 2 is particularly focused on improving the content matching technique for POI retrieval by incorporating word semantics. RQ 2, mentioned in Chapter 1, is reproduced here.

> **RQ 2:** To what extent incorporating semantic association between terms present in POI content, while estimating POI's contextual appropriateness, can improve the contextual POI recommendation quality?

Our experiments (Chapter 4) show that the proposed factored relevance model (FRLM) is effective in matching the content between the POIs in user's preference history and the POIs in the current context by estimating a term weight distribution from both information sources. However, the user profile based RLMs as presented in Chapter 4 can take into account only the document level co-occurrence of terms (ignoring any semantic associations between them). In fact, improvements in the effectiveness of FRLM was not significant specifically with respect to precision at top ranks (nDCG@5, P@5).

Hence to further improve the retrieval effectiveness at top ranks, we incorporate term semantic information into the FRLM in the form of word vector similarities, and propose a word embedding based (estimated with kernel density estimation) further generalized version of factored relevance model, KDE-FRLM (Chapter 5). This leads to a better semantic match between the POI descriptions and the review/description text of the locations visited in the past by a user, which eventually achieves significantly better retrieval performance.

Detailed comparative analysis between FRLM and KDEFRLM reveals that the latter estimates a better term weight distribution for content matching. We observe that the KDE extended version of the factored model mostly outperform its non-semantic (non-KDE) counterpart. This shows that leveraging underlying term semantics of a collection in the form of an embedded space of vectors helps to retrieve more relevant POIs at better ranks.

RQ 2 is hence answered in positive, and we find that the improvements in (KDE)FRLM, by incorporating term semantics, is statistically significant (improvements in nDCG@5, nDCG@10, P@5, and P@10) over a number of IR-based, and RecSys-based baselines. This part of work (along with the multi-contextual generalization) is currently under review in the Information Retrieval Journal [24].

> Anirban Chakraborty, Debasis Ganguly, Annalina Caputo, and Gareth J. F. Jones. Kernel Density Estimation based Factored Relevance Model for Multi-Contextual Point-of-Interest Recommendation. 2021. In *Information Retrieval Journal*, Springer. (Under review). Preprint: `https://arxiv.org/abs/2006.15679`.

Chapter 5 of this thesis is based on this paper.

8.1.3 Multi-Contextual Generalization of FRLM

The third research question RQ 3, mentioned in Chapter 1, is reproduced here.

> **RQ 3:** What is the most effective way to include the *soft* contextual constraints such as 'trip-type', 'accompanied-by' of a given user profile into the POI recommendation framework with a particular focus to improve precision at top ranks?

Initial version of the proposed FRLM or its embedding based variant (KDEFRLM) addresses the (*hard*) location context only, and ignores other non-location type qualifiers (*soft* contextual constraints). In reality, a contextual POI recommender system should also consider a number of non-location type (*soft*) contextual constraints (such as ‘trip type’, ‘accompanied by’) that exist in the *present state* of the user. For example, even if a user’s preference history indicates that the user is an avid beer lover, it may not be suitable to suggest pubs to this user when she is out with her family in the morning.

One major challenge of modeling these *soft* contexts is the inevitable absence of explicit annotation of non-location type context (e.g. trip qualifiers, such as ‘trip purpose’ etc.) in the user preference history (Section 1.4, Chapter 1). While user preference histories generally lack non-location or *trip qualifier*, such information often forms a part of the present state of the user (i.e. the query).

A general approach of bridging this information gap is to employ weak supervision to associate certain topics in user feedback with a seed set of categories defining a precise context, e.g. starting with a seed set of term associations, such as ‘pub’ being relevant to the context category ‘friends’. The natural language text of the reviews is also likely to be helpful in discovering more meaningful dependencies, e.g. associating ‘live music’ with ‘friends’, by using the semantic correlation between ‘pub’ and ‘live music’.

We further generalize the proposed initial framework by introducing multiple contextual constraints. This part contributes to two factors.

Firstly, we incorporate a *generalized framework of addressing both the hard and the soft constraints* (location and trip qualifiers respectively) within the framework of the proposed relevance model. We undertake a weakly supervised approach (leveraging a small set of context-term annotations) to transform the *soft* constraints into term weighting functions.

Further, we incorporate *term semantic information* within the framework of our proposed relevance model. In particular, we use embedded vector representations of words to bridge the vocabulary gap between user preferences, POI descriptions and the trip qualifier (*soft*) constraints.

In Chapter 6 (Section Results), we first observe that including the trip-qualifier based information in the form of joint context (ψ_j) mostly improves the POI retrieval effectiveness. Second, it can be seen that using the *soft* constraint scores as a part of a model is usually more effective than a simple post-hoc combination of these scores with content matching scores. Third, in contrast to a parametric approach, such as SVM, the proposed similarity function ψ_j (Equation 6.2) works better. This is because supervised approaches typically require to rely on large quantities of training data to work well. Moreover, the SVM based approach of [1] did not take into account the semantic similarities between words to estimate the trip-qualifier based appropriateness. It is observed that computing similarities with the embedded word vectors turns out to be more effective.

We would like to mention that both the proposed FRLM and its word embedding based variant (KDEFRLM) have been developed for both the location only (i.e. *hard* context based) retrieval, and the multi-contextual (i.e. *hard+soft*) recommendation. In addition, we also investigate the choice of *different word embedding techniques (in-domain vs. externally trained)* in the effectiveness obtained with our embedding based model KDEFRLM (in both the kernel density estimation process and also in modeling the *soft* contextual constraints).

RQ 3 has been successfully answered, and we conclude that the weakly supervised approach of modeling multiple *soft* constraints further improves the POI recommendation quality. This multi-contextual generalization of the FRLM is published in ACM SIGIR 2020 conference [20].

> Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation. 2020. In *Proceedings of the 43rd International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '20)*, Virtual Conference. Pages 1981 - 1984, ACM, New York, NY, USA.

Chapter 6 of this thesis is based on this paper.

8.1.4 Overall Observations

Key Findings

The key observations from three major contributions of this thesis, i.e. FRLM (Chapter 4), FRLM with word semantics (Chapter 5), and multi-contextual generalization of FRLM (Chapter 6), are stated below.

i) Factored models (*exploitation* and *exploration*) outperform the other approaches.

ii) IR approaches outperform collaborative/personal RecSys ones.

iii) Unsupervised approaches outperform supervised ones.

iv) A combination of (POI) content and (user assigned) tags is more effective than tag-matching alone.

v) Incorporating term semantics improves POI retrieval effectiveness.

vi) Joint context modeling is better for modeling *soft* constraints.

vii) Better precision-oriented and recall-oriented retrieval can be achieved with (factored) relevance modeling.

Key Advantages

We now enlist the key advantages of our proposed approaches for contextual recommendation.

i) Our proposed approaches are *unsupervised* or *weakly supervised.*

The main advantage of unsupervised approaches is that instead of relying on training a model with labelled data, they rather seek to utilize the inherent relationships between latent features of the data itself to make predictions.

ii) Our research is *reproducible*.

Unlike some of the existing approaches that rely on the use of external information (e.g. in the form of Foursquare categories or tags), to overcome reproduciblity and fairness concerns, our experimental setup makes use of a static data collection of POI contents (Section 3.2, Chapter 3).

iii) Our proposed approaches are *suitable for cold-start* recommendation.

As we do not use other users' ratings from external resources, our proposed framework is particularly suitable in an extreme cold-start scenario where no user ratings are available for the candidate POIs, which is a practical problem in many cases [43, 5, 11].

8.1.5 Relevance Feedback with Query Variants

This part of work has a wider scope of research contribution. It is focused on a weakly supervised relevance feedback approach to improve the information retrieval effectiveness in general, which is eventually applied in the specific task of contextual POI recommendation.

In particular, we seek to estimate a robust set of feedback documents by, generally speaking, employing a *document selector function* to decide which documents are useful for relevance feedback. Fourth research question RQ 4, mentioned in Chapter 1, is reproduced here.

> **RQ 4:** To what extent retrievability based document selection for relevance feedback can improve the retrieval effectiveness, specifically with respect to precision at top ranks, both in the general ad-hoc IR setup, and for the specific task of contextual POI recommendation?

In this thesis, we show the effectiveness of IR-based approaches for contextual POI recommendation. In particular, our proposed approaches (Chapter 4, 5, and 6) are based on a pseudo-relevance feedback framework.

In Chapter 7, we argue that standard pseudo-relevance feedback (i.e. PRF) methods have in general been shown to improve overall retrieval effectiveness, such as mean average precision. However, these relevance feedback methods can sometimes, at the cost of increasing recall, lead to decreasing the precision at the very top ranks (e.g. for ranks up to 5).

One of the limitations is that the effectiveness of the PRF algorithms depends, to a large extent, on the choice of this set (say top-M documents), which makes these algorithms less robust and more sensitive to the variations in the chosen set of pseudo-relevant set of documents.

POI recommendation, being a precision-oriented task [1], provides an interesting use-case to study the robustness effects of relevance feedback. Achieving high precision is particularly important from the user's satisfaction perspective, as users are often impatient to scroll down the recommendation list. In addition, real-life use-case often requires that results are to be displayed on mobile devices with limited UI resources.

To mitigate the problem of over-dependence of a pseudo-relevance feedback algorithm on the top-M document set, we make use of a set of equivalence classes of queries rather than one single query. These query equivalents are automatically constructed either from

i) a knowledge base of prior distributions of terms with respect to the given query terms, or

ii) iteratively generated from a relevance model of term distributions in the absence of such priors.

These query variants are then used to estimate the retrievability of each document with the hypothesis that documents that are more likely to be retrieved at top-ranks for a larger number of these query variants are more likely to be effective for relevance feedback.

Results of our experiments (Chapter 7) show that our proposed method of a weakly supervised relevance model, WSRM (and WSFRM, which is a counter part of our factored model, FRLM) is able to achieve substantially better precision at top-ranks (e.g. higher nDCG@5 and P@5 values) for ad-hoc IR and points-of-interest (POI) recommendation tasks.

For TREC ad-hoc, although traditional relevance model (RM) improves MAP substantially for each topic set, it is seen that the improvements in 'P@5' are marginal. It is also observed that the kNN-based re-sampling method is able to substantially improve precision at top ranks (as compared to RM). However, our approach generalizes well on the test data indicates that WSRM is more resilient to parameter variation effects. Somewhat to our surprise, we observe that fusion based approaches tend to work well with manually annotated query variants, when each query variant points to the exact same information need.

We investigate the differences between the two feedback document sets, top-M (i.e. the standard set of pseudo relevant documents) and top-M' (i.e. the set of documents selected based on their retrievability scores) as obtained by the three feedback document selection methods, namely kNN, WSRM and CWSRM (Section 7.6.1, Chapter 7). These differences are measured at a number of different rank cut-off points. We observe that our approach achieves the desired trade-off between exploration (i.e. leveraging information from new documents) and exploitation (i.e. making use of the top-M set).

RQ 4 is hence answered in positive, and we observe that our proposed approach consistently improves precision at top ranks (e.g. higher nDCG@5 and P@5 values) in two different tasks, namely TREC ad-hoc and the contextual POI recommendation. This work of relevance feedback with query variants is published in ACM CIKM 2020 conference [21].

> Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. Retrievability based Document Selection for Relevance Feedback with Automatically Generated Query Variants. 2020. In *29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, Virtual Conference. Pages 125 - 134, ACM, New York, NY, USA.

Chapter 7 of this thesis is based on this paper.

8.2 Future Work

We now discuss a number of potential directions as future work that may be done along this line of research.

8.2.1 More Fine-grained Location Context for Contextual Recommendation

In our experimental setup, the location constraint is *hard*, and the system must make recommendations for the current location (city) only because a POI in a different location (city) is obviously non-relevant. One may argue that the location context can also be a *soft* constraint, where accurate geo-coordinates can be taken into consideration for favouring POIs that are in close proximity of the user's accurate coordinates [105]. However, addressing this is beyond the scope of this thesis and is a potential future work, possibly involving simulated users within the geographical bounding box of a city.

8.2.2 More (*Soft*) Contexts for Contextual Recommendation

According to our experimental setup, the current context of a user forms a part of the query comprised of a pair of trip qualifiers of the form (L, Q), where L is the location (*hard*) context, and $Q = Q_1 \times \ldots Q_c$ is a combination of $c = 3$ non-location (*soft*) contexts. In particular, Q_1=`trip-type`, e.g. vacation, Q_2=`trip-duration`, e.g. day-trip, and Q_3=`accompanied-by`, e.g. solo or with friends (Secion 3.1.1, Chapter 3).

Indeed, in a general case it should be possible to include a number of contextual constraints such as geographical influence [95], time of the day [97], road traffic or availability of transportation, current weather etc. as a part of the non-location type constraints (i.e. use a value of c higher than that of 3). However, we restrict the scope of our investigation to three specific non-location type attributes only and leave the other attributes for a possible future extension of this work.

In future, we aim to extend our experiments to include additional information as a part of a user's context, e.g. the fine-grained location of a user in terms of GPS coordinates (instead of simply a city name), environmental context (e.g. if a user is indoors or outdoors), traveling amenities context (e.g. if the user has private transport) etc. One possible way to obtain such additional contextual information would be to apply simulation techniques seeking to model the travel behaviour of simulated user agents.

8.2.3 Learning Per User-Profile *Exploitation-Exploration* Trade-off

Our proposed model for contextual recommendation is essentially a two-step factored relevance model. It estimates a relevance model based on the user's preference history first. Then it estimates another relevance model based on both the initial model and the top retrieved POIs in the current context. Finally these two models are linearly combined.

Instead of a relatively simple approach of employing a constant value for the linear combination parameter γ_H, we investigate if individually choosing the values of this parameter based on the user profiles (queries) can lead to better results (Section 6.5.4, Chapter 6). In particular, we conduct a grid-based exploration of the parameter γ_H for each query separately.

We observe that different queries achieve optimal results with different values of the exploration-exploitation parameter. This in turn indicates that for some user profiles it is better to rely to a greater degree on the historical preferences (*exploitation*) whereas for some other ones it is better to allow provision for more *exploration* into the POI descriptors. Our results also suggests that automatically estimating the value of the exploration-exploitation trade-off can potentially improve results further. This we leave as a future exercise.

8.2.4 Query Variants for Verbose Query

We proposed a concept of weakly supervised relevance models by using the notion of *retrievability* from automatically constructed query variants to improve the quality of relevance feedback. We observed that our approach consistently improves precision at top ranks in two different tasks, namely TREC ad-hoc and the contextual POI recommendation. However, standard queries (such as TREC ad-hoc queries) are usually short and precise in indicating the actual information need. Standard IR approaches do not perform well for longer or verbose queries [41, 47, 12].

As a future exercise, we would like to investigate how effectively can one generate the query variants for verbose queries, and how effective will these verbose variants be for improving retrieval effectiveness.

Bibliography

[1] Mohammad Aliannejadi and Fabio Crestani. "Personalized Context-Aware Point of Interest Recommendation". In: *ACM Trans. Inf. Syst.* 36.4 (Oct. 2018), 45:1–45:28. ISSN: 1046-8188. DOI: 10.1145/3231933. URL: http://doi.acm.org/10.1145/3231933.

[2] Mohammad Aliannejadi and Fabio Crestani. "Venue Appropriateness Prediction for Personalized Context-Aware Venue Suggestion". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: ACM, 2017, pp. 1177–1180. ISBN: 978-1-4503-5022-8. DOI: 10.1145/3077136.3080754. URL: http://doi.acm.org/10.1145/3077136.3080754.

[3] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. "A Cross-Platform Collection for Contextual Suggestion". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: ACM, 2017, pp. 1269–1272. ISBN: 978-1-4503-5022-8. DOI: 10.1145/3077136.3080752. URL: http://doi.acm.org/10.1145/3077136.3080752.

[4] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. "Personalized keyword boosting for venue suggestion based on multiple LBSNs". In: *European Conference on Information Retrieval*. Springer. 2017, pp. 291–303.

[5] Avi Arampatzis and Georgios Kalamatianos. "Suggesting Points-of-Interest via Content-Based, Collaborative, and Hybrid Fusion Methods in Mobile Devices". In: *ACM Trans. Inf. Syst.* 36.3 (Sept. 2017). ISSN: 1046-8188. DOI: 10.1145/3125620. URL: https://doi.org/10.1145/3125620.

[6] Leif Azzopardi and Vishwa Vinay. "Retrievability: An Evaluation Measure for Higher Order Information Access Tasks". In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM '08. Napa Valley, California, USA: Association for Computing Machinery, 2008, pp. 561–570. ISBN: 9781595939913. DOI: 10.1145/1458082.1458157. URL: https://doi.org/10.1145/1458082.1458157.

[7] Peter Bailey et al. "Retrieval Consistency in the Presence of Query Variations". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: Association for Computing Machinery, 2017,

pp. 395–404. ISBN: 9781450350228. DOI: 10.1145/3077136.3080839. URL: https://doi.org/10.1145/3077136.3080839.

[8] Peter Bailey et al. “UQV100: A Test Collection with Query Variability”. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’16. Pisa, Italy: Association for Computing Machinery, 2016, pp. 725–728. ISBN: 9781450340694. DOI: 10.1145/2911451.2914671. URL: https://doi.org/10.1145/2911451.2914671.

[9] Shariq Bashir and Andreas Rauber. “Improving Retrievability of Patents with Cluster-Based Pseudo-Relevance Feedback Documents Selection”. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM ’09. Hong Kong, China: Association for Computing Machinery, 2009, pp. 1863–1866. ISBN: 9781605585123. DOI: 10.1145/1645953.1646250. URL: https://doi.org/10.1145/1645953.1646250.

[10] Mostafa Bayomi and Séamus Lawless. “ADAPT_TCD: An Ontology-Based Context Aware Approach for Contextual Suggestion”. In: *TREC 2016*. 2016.

[11] Mostafa Bayomi et al. “CoRE: A Cold-start Resistant and Extensible Recommender System”. In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC ’19. Limassol, Cyprus: ACM, 2019, pp. 1679–1682. ISBN: 978-1-4503-5933-7. DOI: 10.1145/3297280.3297601. URL: http://doi.acm.org/10.1145/3297280.3297601.

[12] Michael Bendersky and W. Bruce Croft. “Discovering Key Concepts in Verbose Queries”. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’08. Singapore, Singapore: Association for Computing Machinery, 2008, pp. 491–498. ISBN: 9781605581644. DOI: 10.1145/1390334.1390419. URL: https://doi.org/10.1145/1390334.1390419.

[13] Rodger Benham et al. “Boosting Search Performance Using Query Variations”. In: *ACM Trans. Inf. Syst.* 37.4 (Oct. 2019). ISSN: 1046-8188. DOI: 10.1145/3345001. URL: https://doi.org/10.1145/3345001.

[14] Rodger Benham et al. “Towards Efficient and Effective Query Variant Generation”. In: *Proc. of DESIRES ’18*. 2018, pp. 62–67.

[15] Bodo Billerbeck and Justin Zobel. “Questioning Query Expansion: An Examination of Behaviour and Parameters”. In: *Proceedings of the 15th Australasian Database Conference - Volume 27*. ADC ’04. Dunedin, New Zealand: Australian Computer Society, Inc., 2004, pp. 69–76.

[16] Guihong Cao et al. “Selecting Good Expansion Terms for Pseudo-Relevance Feedback”. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’08. Singapore, Singapore: Association for Computing Machinery, 2008, pp. 243–250. ISBN: 9781605581644. DOI: 10.1145/1390334.1390377. URL: https://doi.org/10.1145/1390334.1390377.

[17] Ben Carterette et al. “Overview of the TREC 2014 Session Track”. In: *Proc. of TREC 2014*. 2014.

[18] Anirban Chakraborty. “Enhanced Contextual Recommendation Using Social Media Data”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR ’18. Ann Arbor, MI, USA: ACM, 2018, pp. 1455–1455. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3210223. URL: http://doi.acm.org/10.1145/3209978.3210223.

[19] Anirban Chakraborty. “Exploring Search Behaviour in Microblogs”. In: *Seventh BCS-IRSG Symposium on Future Directions in Information Access, FDIA 2017, 5 September 2017, Barcelona, Spain*. 2017. DOI: 10.14236/ewic/FDIA2017.8. URL: https://doi.org/10.14236/ewic/FDIA2017.8.

[20] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. “Relevance Models for Multi-Contextual Appropriateness in Point-of-Interest Recommendation”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 1981–1984. ISBN: 9781450380164. DOI: 10.1145/3397271.3401197. URL: https://doi.org/10.1145/3397271.3401197.

[21] Anirban Chakraborty, Debasis Ganguly, and Owen Conlan. “Retrievability Based Document Selection for Relevance Feedback with Automatically Generated Query Variants”. In: *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. CIKM ’20. Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 125–134. ISBN: 9781450368599. DOI: 10.1145/3340531.3412032. URL: https://doi.org/10.1145/3340531.3412032.

[22] Anirban Chakraborty, Kripabandhu Ghosh, and Swapan Kumar Parui. “Retrieval from Noisy E-Discovery Corpus in the Absence of Training Data”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 755–758. ISBN: 9781450336215. DOI: 10.1145/2766462.2767828. URL: https://doi.org/10.1145/2766462.2767828.

[23] Anirban Chakraborty et al. “A Factored Relevance Model for Contextual Point-of-Interest Recommendation”. In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ICTIR ’19. Santa Clara, CA, USA: ACM, 2019, pp. 157–164. ISBN: 978-1-4503-6881-0. DOI: 10.1145/3341981.3344230. URL: http://doi.acm.org/10.1145/3341981.3344230.

[24] Anirban Chakraborty et al. “Kernel Density Estimation based Factored Relevance Model for Multi-Contextual Point-of-Interest Recommendation”. In: *Information Retrieval Journal* (2021), Under review. Preprint available at https://arxiv.org/abs/2006.15679.

[25] Li Chen, Guanliang Chen, and Feng Wang. “Recommender systems based on user reviews: the state of the art”. In: *User Modeling and User-Adapted Interaction* 25.2 (2015), pp. 99–154.

[26] Chen Cheng et al. “Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks”. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI ’12. Toronto, Ontario, Canada: AAAI Press, 2012, pp. 17–23.

[27] Cyril Cleverdon. “The Cranfield Tests on Index Language Devices”. In: *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 47–59. ISBN: 1558604545.

[28] W. B. Croft. “Knowledge-based and statistical approaches to text retrieval”. In: *IEEE Expert* 8.2 (1993), pp. 8–12.

[29] Adriel Dean-Hall et al. “Overview of the TREC 2012 Contextual Suggestion Track”. In: *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*. 2012. URL: http://trec.nist.gov/pubs/trec21/papers/CONTEXTUAL12.overview.pdf.

[30] Romain Deveaud et al. “Experiments with a Venue-Centric Model for Personalisedand Time-Aware Venue Suggestion”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM ’15. Melbourne, Australia: ACM, 2015, pp. 53–62. ISBN: 978-1-4503-3794-6. DOI: 10.1145/2806416.2806484. URL: http://doi.acm.org/10.1145/2806416.2806484.

[31] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[32] Quan Fang et al. “STCAPLRS: A Spatial-Temporal Context-Aware Personalized Location Recommendation System”. In: *ACM Trans. Intell. Syst. Technol.* 7.4 (Mar. 2016), 59:1–59:30. ISSN: 2157-6904. DOI: 10.1145/2842631. URL: http://doi.acm.org/10.1145/2842631.

[33] G. W. Furnas et al. “The Vocabulary Problem in Human-System Communication”. In: *Commun. ACM* 30.11 (Nov. 1987), pp. 964–971. ISSN: 0001-0782. DOI: 10.1145/32206.32212. URL: https://doi.org/10.1145/32206.32212.

[34] Debasis Ganguly et al. “Word Embedding Based Generalized Language Model for Information Retrieval”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 795–798. ISBN: 9781450336215. DOI: 10.1145/2766462.2767780. URL: https://doi.org/10.1145/2766462.2767780.

[35] Huiji Gao et al. “Exploring Temporal Effects for Location Recommendation on Location-based Social Networks”. In: *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys ’13. Hong Kong, China: ACM, 2013, pp. 93–100. ISBN: 978-1-4503-2409-0. DOI: 10.1145/2507157.2507182. URL: http://doi.acm.org/10.1145/2507157.2507182.

[36] Rainer Gemulla et al. “Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’11. San Diego, California, USA: Association for Computing Machinery, 2011, pp. 69–77. ISBN: 9781450308137. DOI: 10.1145/2020408.2020426. URL: https://doi.org/10.1145/2020408.2020426.

[37] Kripabandhu Ghosh, Anirban Chakraborty, and Swapan Kumar Parui. “Improving IR Performance from OCRed Text Using Cooccurrence”. In: *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*. FIRE ’12 & ’13. New Delhi, India: Association for Computing Machinery, 2013. ISBN: 9781450328302. DOI: 10.1145/2701336.2701648. URL: https://doi.org/10.1145/2701336.2701648.

[38] Kripabandhu Ghosh et al. “Improving Information Retrieval Performance on OCRed Text in the Absence of Clean Text Ground Truth”. In: *Information Processing & Management* 52.5 (2016), pp. 873–884. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2016.03.006. URL: http://www.sciencedirect.com/science/article/pii/S030645731630036X.

[39] Jean-Benoit Griesner, Talel Abdessalem, and Hubert Naacke. “POI Recommendation: Towards Fused Matrix Factorization with Geographical and Temporal Influences”. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys ’15. Vienna, Austria: ACM, 2015, pp. 301–304. ISBN: 978-1-4503-3692-5. DOI: 10.1145/2792838.2799679. URL: http://doi.acm.org/10.1145/2792838.2799679.

[40] Jiafeng Guo et al. “A Deep Relevance Matching Model for Ad-Hoc Retrieval”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM ’16. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016, pp. 55–64. ISBN: 9781450340731. DOI: 10.1145/2983323.2983769. URL: https://doi.org/10.1145/2983323.2983769.

[41] Manish Gupta and Michael Bendersky. “Information Retrieval with Verbose Queries”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 1121–1124. ISBN: 9781450336215. DOI: 10.1145/2766462.2767877. URL: https://doi.org/10.1145/2766462.2767877.

[42] Donna Harman. “Overview of the fourth text retrieval conference (TREC-4)”. In: *NIST Special Publication* 500236 (1996), pp. 1–23.

[43] Seyyed Hadi Hashemi et al. “Overview of the TREC 2016 contextual suggestion track”. In: *Proceedings of TREC*. Vol. 2016. 2016.

[44] Ben He and Iadh Ounis. “Finding Good Feedback Documents”. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM ’09. Hong Kong, China: Association for Computing Machinery, 2009, pp. 2011–2014. ISBN: 9781605585123. DOI: 10.1145/1645953.1646289. URL: https://doi.org/10.1145/1645953.1646289.

[45] Xiangnan He et al. "Neural Collaborative Filtering". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 173–182. ISBN: 978-1-4503-4913-0. DOI: 10.1145/3038912.3052569. URL: https://doi.org/10.1145/3038912.3052569.

[46] Djoerd Hiemstra. *Using language models for information retrieval*. PhD thesis, Center of Telematics and Information Technology, AE Enschede, 2000.

[47] Samuel Huston and W. Bruce Croft. "Evaluating Verbose Query Processing Techniques". In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '10. Geneva, Switzerland: Association for Computing Machinery, 2010, pp. 291–298. ISBN: 9781450301534. DOI: 10.1145/1835449.1835499. URL: https://doi.org/10.1145/1835449.1835499.

[48] Nasreen Abdul Jaleel et al. "UMass at TREC 2004: Novelty and HARD". In: *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*. 2004. URL: http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf.

[49] Ming Jiang and Daqing He. "PITT at TREC 2013 Contextual Suggestion Track". In: *TREC 2013*. 2013.

[50] K. Sparck Jones, S. Walker, and S. E. Robertson. "A probabilistic model of information retrieval: development and comparative experiments". In: *Information Processing and Management*. 2000, pp. 779–840.

[51] Victor Lavrenko and W. Bruce Croft. "Relevance Based Language Models". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New Orleans, Louisiana, USA: ACM, 2001, pp. 120–127. ISBN: 1-58113-331-6. DOI: 10.1145/383952.383972. URL: http://doi.acm.org/10.1145/383952.383972.

[52] Kyung Soon Lee, W. Bruce Croft, and James Allan. "A Cluster-Based Resampling Method for Pseudo-Relevance Feedback". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '08. Singapore, Singapore: Association for Computing Machinery, 2008, pp. 235–242. ISBN: 9781605581644. DOI: 10.1145/1390334.1390376. URL: https://doi.org/10.1145/1390334.1390376.

[53] Asher Levi et al. "Finding a Needle in a Haystack of Reviews: Cold Start Context-based Hotel Recommender System". In: *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys '12. Dublin, Ireland: ACM, 2012, pp. 115–122. ISBN: 978-1-4503-1270-7. DOI: 10.1145/2365952.2365977. URL: http://doi.acm.org/10.1145/2365952.2365977.

[54] Hanchen Li et al. "BJUT at TREC 2014 contextual suggestion track: Hybrid recommendation based on open-web information". In: *TREC 2014*. 2014.

[55] Hua Li and Rafael Alonso. "User modeling for contextual suggestion". In: *TREC 2014*. 2014.

[56] Xin Li et al. "Next and Next New POI Recommendation via Latent Behavior Pattern Inference". In: *ACM Trans. Inf. Syst.* 37.4 (Sept. 2019). ISSN: 1046-8188. DOI: `10.1145/3354187`. URL: `https://doi.org/10.1145/3354187`.

[57] Jimmy Lin. "The neural hype and comparisons against weak baselines". In: *ACM SIGIR Forum*. Vol. 52. 2. ACM New York, NY, USA. 2019, pp. 40–51.

[58] Binsheng Liu et al. "A Comparative Analysis of Human and Automatic Query Variants". In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ICTIR '19. Santa Clara, CA, USA: Association for Computing Machinery, 2019, pp. 47–50. ISBN: 9781450368810. DOI: `10.1145/3341981.3344223`. URL: `https://doi.org/10.1145/3341981.3344223`.

[59] Bin Liu et al. "Learning Geographical Preferences for Point-of-Interest Recommendation". In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Chicago, Illinois, USA: Association for Computing Machinery, 2013, pp. 1043–1051. ISBN: 9781450321747. DOI: `10.1145/2487575.2487673`. URL: `https://doi.org/10.1145/2487575.2487673`.

[60] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[61] Xiaolu Lu et al. "Relevance Modeling with Multiple Query Variations". In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ICTIR '19. Santa Clara, CA, USA: Association for Computing Machinery, 2019, pp. 27–34. ISBN: 9781450368810. DOI: `10.1145/3341981.3344224`. URL: `https://doi.org/10.1145/3341981.3344224`.

[62] Yuanhua Lv and ChengXiang Zhai. "A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback". In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. Hong Kong, China: ACM, 2009, pp. 1895–1898. ISBN: 978-1-60558-512-3. DOI: `10.1145/1645953.1646259`. URL: `http://doi.acm.org/10.1145/1645953.1646259`.

[63] Hao Ma et al. "Improving Recommender Systems by Incorporating Social Contextual Information". In: *ACM Trans. Inf. Syst.* 29.2 (Apr. 2011). ISSN: 1046-8188. DOI: `10.1145/1961209.1961212`. URL: `https://doi.org/10.1145/1961209.1961212`.

[64] Hao Ma et al. "SoRec: Social Recommendation Using Probabilistic Matrix Factorization". In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM '08. Napa Valley, California, USA: Association for Computing Machinery, 2008, pp. 931–940. ISBN: 9781595939913. DOI: 10.1145/1458082.1458205. URL: https://doi.org/10.1145/1458082.1458205.

[65] Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge university press, 2008.

[66] Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. "Relevance feedback and query expansion". In: *Introduction to information retrieval*. Cambridge university press, 2008, pp. 177–194.

[67] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. "Modelling user preferences using word embeddings for context-aware venue recommendation". In: *arXiv preprint arXiv:1606.07828* (2016).

[68] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119. URL: http://dl.acm.org/citation.cfm?id=2999792.2999959.

[69] Koji Miyahara and Michael J. Pazzani. "Collaborative Filtering with the Simple Bayesian Classifier". In: *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*. PRICAI '00. Melbourne, Australia: Springer-Verlag, 2000, pp. 679–689. ISBN: 3540679251.

[70] Ali Montazeralghaem, Hamed Zamani, and James Allan. "A Reinforcement Learning Framework for Relevance Feedback". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 59–68. ISBN: 9781450380164. DOI: 10.1145/3397271.3401099. URL: https://doi.org/10.1145/3397271.3401099.

[71] Claudiu-Cristian Musat, Yizhong Liang, and Boi Faltings. "Recommendation Using Textual Opinions". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, 2013, pp. 2684–2690. ISBN: 9781577356332.

[72] Paul Ogilvie, Ellen Voorhees, and Jamie Callan. "On the number of terms used in automatic query expansion". In: *Information Retrieval* 12.6 (2009), pp. 666–679.

[73] Jiaul H. Paik. "A Probabilistic Model for Information Retrieval Based on Maximum Value Distribution". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 585–594. ISBN: 9781450336215. DOI: 10.1145/2766462.2767762. URL: https://doi.org/10.1145/2766462.2767762.

[74] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[75] Jay M. Ponte and W. Bruce Croft. “A Language Modeling Approach to Information Retrieval”. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’98. Melbourne, Australia: Association for Computing Machinery, 1998, pp. 275–281. ISBN: 1581130155. DOI: 10.1145/290941.291008. URL: https://doi.org/10.1145/290941.291008.

[76] Francesco Ricci et al. *Recommender Systems Handbook*. Springer US, 2011.

[77] S. E. Robertson and S. Walker. “Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval”. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’94. Dublin, Ireland: Springer-Verlag, 1994, pp. 232–241. ISBN: 038719889X.

[78] S.E. Robertson et al. *Okapi at TREC-4*. 1996.

[79] Stephen E Robertson. “The probability ranking principle in IR”. In: *Journal of Documentation* (1977).

[80] Stephen E Robertson et al. “Okapi at TREC-3”. In: *Nist Special Publication Sp* 109 (1995), p. 109.

[81] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019.

[82] Dwaipayan Roy, Ayan Bandyopadhyay, and Mandar Mitra. “A Simple Context Dependent Suggestion System.” In: *TREC 2013*. 2013.

[83] Dwaipayan Roy et al. “Using Word Embeddings for Information Retrieval: How Collection and Term Normalization Choices Affect Performance”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM ’18. Torino, Italy: Association for Computing Machinery, 2018, pp. 1835–1838. ISBN: 9781450360142. DOI: 10.1145/3269206.3269277. URL: https://doi.org/10.1145/3269206.3269277.

[84] Dwaipayan Roy et al. “Word Vector Compositionality Based Relevance Feedback Using Kernel Density Estimation”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM ’16. Indianapolis, Indiana, USA: ACM, 2016, pp. 1281–1290. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983750. URL: http://doi.acm.org/10.1145/2983323.2983750.

[85] Ruslan Salakhutdinov and Andriy Mnih. “Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Helsinki, Finland: ACM, 2008, pp. 880–887. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390267. URL: http://doi.acm.org/10.1145/1390156.1390267.

[86] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. USA: Prentice-Hall, Inc., 1971.

[87] Procheta Sen, Debasis Ganguly, and Gareth Jones. “Word-Node2Vec: Improving Word Embedding with Document-Level Non-Local Word Co-occurrences”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1041–1051. DOI: `10.18653/v1/N19-1109`. URL: `https://www.aclweb.org/anthology/N19-1109`.

[88] Joseph A. Shaw and Edward A. Fox. “Combination of Multiple Searches”. In: *The Second Text REtrieval Conference (TREC-2)*. 1994, pp. 243–252.

[89] Harald Steck. “Item Popularity and Recommendation Accuracy”. In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys ’11. Chicago, Illinois, USA: Association for Computing Machinery, 2011, pp. 125–132. ISBN: 9781450306836. DOI: `10.1145/2043932.2043957`. URL: `https://doi.org/10.1145/2043932.2043957`.

[90] Alessandro Suglia et al. “A Deep Architecture for Content-Based Recommendations Exploiting Recurrent Neural Networks”. In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. UMAP ’17. Bratislava, Slovakia: Association for Computing Machinery, 2017, pp. 202–211. ISBN: 9781450346351. DOI: `10.1145/3079628.3079684`. URL: `https://doi.org/10.1145/3079628.3079684`.

[91] Egidio Terra and Robert Warren. “Poison Pills: Harmful Relevant Documents in Feedback”. In: *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. CIKM ’05. Bremen, Germany: Association for Computing Machinery, 2005, pp. 319–320. ISBN: 1595931406. DOI: `10.1145/1099554.1099646`. URL: `https://doi.org/10.1145/1099554.1099646`.

[92] E. Voorhees and D. Harman. “Overview of the Eighth Text REtrieval Conference (TREC-8)”. In: *TREC*. 1999.

[93] Ellen Voorhees. “Overview of the TREC 2004 Robust Track”. In: *TREC*. 2004.

[94] Peilin Yang and Hui Fang. “An exploration of ranking-based strategy for contextual suggestion”. In: *TREC 2012*. 2012.

[95] Mao Ye et al. “Exploiting Geographical Influence for Collaborative Point-of-interest Recommendation”. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’11. Beijing, China: ACM, 2011, pp. 325–334. ISBN: 978-1-4503-0757-4. DOI: `10.1145/2009916.2009962`. URL: `http://doi.acm.org/10.1145/2009916.2009962`.

[96] Yonghong Yu and Xingguo Chen. “A survey of point-of-interest recommendation in location-based social networks”. In: *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[97] Quan Yuan et al. “Time-aware Point-of-interest Recommendation”. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR ’13. Dublin, Ireland: ACM, 2013, pp. 363–372. ISBN: 978-1-4503-2034-4. DOI: 10.1145/2484028.2484030. URL: http://doi.acm.org/10.1145/2484028.2484030.

[98] Hamed Zamani and W. Bruce Croft. “Embedding-Based Query Language Models”. In: *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval.* ICTIR ’16. Newark, Delaware, USA: Association for Computing Machinery, 2016, pp. 147–156. ISBN: 9781450344975. DOI: 10.1145/2970398.2970405. URL: https://doi.org/10.1145/2970398.2970405.

[99] Hamed Zamani and W. Bruce Croft. “Relevance-Based Word Embedding”. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR ’17. Shinjuku, Tokyo, Japan: Association for Computing Machinery, 2017, pp. 505–514. ISBN: 9781450350228. DOI: 10.1145/3077136.3080831. URL: https://doi.org/10.1145/3077136.3080831.

[100] Hamed Zamani et al. “Pseudo-Relevance Feedback Based on Matrix Factorization”. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* CIKM ’16. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016, pp. 1483–1492. ISBN: 9781450340731. DOI: 10.1145/2983323.2983844. URL: https://doi.org/10.1145/2983323.2983844.

[101] Oleg Zendel et al. “Information Needs, Queries, and Query Performance Prediction”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR’19. Paris, France: Association for Computing Machinery, 2019, pp. 395–404. ISBN: 9781450361729. DOI: 10.1145/3331184.3331253. URL: https://doi.org/10.1145/3331184.3331253.

[102] ChengXiang Zhai. “Statistical language models for information retrieval”. In: *Synthesis lectures on human language technologies* 1.1 (2008), pp. 1–141.

[103] Chengxiang Zhai and John Lafferty. “A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval”. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR ’01. New Orleans, Louisiana, USA: Association for Computing Machinery, 2001, pp. 334–342. ISBN: 1581133316. DOI: 10.1145/383952.384019. URL: https://doi.org/10.1145/383952.384019.

[104] Chengxiang Zhai and John Lafferty. “A Study of Smoothing Methods for Language Models Applied to Information Retrieval”. In: *ACM Trans. Inf. Syst.* 22.2 (Apr. 2004), pp. 179–214. ISSN: 1046-8188. DOI: 10.1145/984321.984322. URL: https://doi.org/10.1145/984321.984322.

[105] Jia-Dong Zhang and Chi-Yin Chow. “GeoSoCa: Exploiting Geographical, Social and Categorical Correlations for Point-of-Interest Recommendations”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 443–452. ISBN: 9781450336215. DOI: `10.1145/2766462.2767711`. URL: `https://doi.org/10.1145/2766462.2767711`.

[106] Guoqing Zheng and Jamie Callan. “Learning to Reweight Terms with Distributed Representations”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: Association for Computing Machinery, 2015, pp. 575–584. ISBN: 9781450336215. DOI: `10.1145/2766462.2767700`. URL: `https://doi.org/10.1145/2766462.2767700`.