Inferring User Needs & Tasks from User Interactions

Rishabh Mehrotra

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

April 19, 2018

I, Rishabh Mehrotra, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

The need for search often arises from a broad range of complex information needs or tasks (such as booking travel, buying a house, etc.) which lead to lengthy search processes characterised by distinct stages and goals. While existing search systems are adept at handling simple information needs, they offer limited support for tackling complex tasks. Accurate task representations could be useful in aptly placing users in the task-subtask space and enable systems to contextually target the user, provide them better query suggestions, personalization and recommendations and help in gauging satisfaction.

The major focus of this thesis is to work towards task based information retrieval systems - search systems which are adept at understanding, identifying and extracting tasks as well as supporting user's complex search task missions. This thesis focuses on two major themes: (i) developing efficient algorithms for understanding and extracting search tasks from log user and (ii) leveraging the extracted task information to better serve the user via different applications. Based on log analysis on a tera-byte scale data from a real-world search engine, detailed analysis is provided on user interactions with search engines. On the task extraction side, two bayesian non-parametric methods are proposed to extract subtasks from a complex task and to recursively extract hierarchies of tasks and subtasks. A novel coupled matrix-tensor factorization model is proposed that represents user based on their topical interests and task behaviours.

Beyond personalization, the thesis demonstrates that task information provides better context to learn from and proposes a novel neural task context embedding architecture to learn query representations. Finally, the thesis examines implicit sig-

nals of user interactions and considers the problem of predicting user's satisfaction when engaged in complex search tasks. A unified multi-view deep sequential model is proposed to make query and task level satisfaction prediction.

Acknowledgements

Almost eight years ago, during the winters of 2009 I contacted a professor for an internship project after the second year of my undergraduate studies. It was the start of a fruitful period during which I learned as much about the topic of my research and science in general as I did about the people around me and myself. This period now comes to an end and I want to thank all the people that played an important role.

I am indebted at my supervisor Emine Yilmaz for offering me the chance to pursue PhD research at UCL. I cannot thank Emine enough. She immensely helped me shape my research career as it stands today. She taught me insights about doing research, asking the right questions, emphasizing the role of evaluation and metrics, and how to best answer research questions. Her teachings, however, went far beyond science when we spent much of our weekly meetings talking about much broader topics than the papers we were writing. Thank you Emine for being an inspirational guide and friend along my PhD journey.

I wouldn't have had a chance to pursue a PhD had it not been for the opportunity offered by Scott Sanner, who back in 2010 helped me start my research career. Scott has been inspirational in more ways than I can count. From supporting me in my early research endeavours, to helping me attend summer schools, and giving me the chance of working on my first major Machine Learning project at his group at NICTA, Scott has played an instrumental role in laying the building blocks of my research career.

I am honored to have Shi Zhou and Fabrizio Silvestri serving on my PhD committee and appreciate their thoughtful comments and feedback on my thesis. I'm

thankful to Thore Graepel, Filip Radlinksi, Sebastian Riedel and Ingemar Cox for serving on my first year and transfer viva committees.

I also want to thank other members of the UCL family, including my secondary supervisor Ingemar Cox, my labmates Manisha, Shangsong, Chris, Tim and Jiyin, who made my time at UCL enjoyable. I also thank other UCL research colleagues Sebastian Reidel, Jun Wang, Weinan Zhang, Marc Sloan, Marzieh, Ivan, Andreas, Bin Zhou, and Jagadeesh for their kind help and insightful discussions. I would like to especially thank the excellent administrative staff at UCL, including the former and current research administrators Melanie Johnson and Sarah Turnbull for providing a nurturing support system to the students of our department, to Dawn Bailey and Wendy Richards for helping with the finances and enabling us to attend conferences, and to JJ Giwa for being the awesome superhuman manager handling just about everything at our CS department.

I spent many months interning with amazing researchers and colleagues at Microsoft Research and Bing. My internships gave me the opportunity to learn from and work with Susan Dumais, Paul Bennett, Jaime Teevan, Katja Hofmann, Abhishek Arun, Fernando Diaz, Milad Shokouhi, Hanna Wallach, Ahmed Hassan, Imed Zitouni, Ahmed El Kohly, Madian Khabsa, Amit Sharma and Ashton Anderson. I enjoyed a lot working along side them, and co-authoring papers with them. Working with Sue, Paul and Jaime taught me a lot about analytical thinking, while Katja and Abhishek helped me develop experimentation skills and got me started thinking about counterfactual evaluation. The team at MSR NYC has been one of the most amazing collaborations experience I have ever had, with Fernando, Hanna, Amit and Ashton bringing diverse perspectives into discussions. Collaborations with Milad, Ahmed Hassan and Imed while I was at Bellevue have also been very valuable to me, and working with Ahmed on CIKM tutorial and WSDM workshop has been exciting.

I wish to thank Milad Shokouhi for being a mentor, a friend and a guide. I cherish the long insightful discussions I've had with him about all matters, personal and professional alike. Thanks Milad for looking out for me!

I want to thank my co-organizers of the TREC Tasks Track Emine, Manisha, Ben, Evangelos, Nick and Peter for taking me on board for the TREC initiative. I want to thank the research community, of which I feel I have become part, for being awesome. It was great to meet all these wonderful people again and again wherever there was a conference.

Special thanks to Chetan Mehrotra for being an inspirational brother right from my school days and to my friends back home and at BITS: Aqueel, Agrita, Moin, Psalm, Shitanshu, Rushabh, Mayank, Aman, Saurabh, Vivek Jain, Pranav, Shrainik, and BBT for the long discussions and friendly chats. I would also like to thank Prasanta Bhattacharya, who has helped me countless times right from my undergrad days and who still continues to be a go-to person for insightful discussions on any topic. My London flatmates and brother Deval and friends Inder, Aakash for the enjoyable time and eventful nights in London.

The path towards my PhD wasn't always as cheerful as I would have liked, and I wish to thank Abhay Kacker for providing support during tough times, and to Shubhay Kacker for being his adorable self, bringing smiles to all our faces.

Last but not least, I am very grateful to my parents and my sister for their love and unconditional support during these years. I am thankful to them for shielding me from the vicissitudes of life, for teaching me the values of patience, hard work and honestly, and for providing me a nurturing environment wherein I could focus on learning. All I am I am because of their support, sacrifices and encouragement throughout my life. Everything I am, and everything I will ever be, will be because of them.

Dedicated to my parents, Renu & Ajai Mehrotra, my sister Shruti & to Shubhay!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Search Engines essentially act as filters for the wealth of information available on the Internet. They allow users to quickly and easily find information that is of genuine interest or value to them, without the need to wade through numerous irrelevant web pages. Over six billion web searches are performed every single day [1] by over half the world's population. Given the democratization of content creation via the internet, the number of web pages indexed by search engines have increased from 1 billion in 2000 to over 60 trillion in 2017 [2] equivalent of multi-petabytes of searchable data potentially consumable by internet users.

In order to prevent internet users from drowning in sea of irrelevant information and irrelevant marketing messages, search engines handle the bulk of information filtering for the users. For many people, web search engines such as Baidu, Bing, Google, and Yandex are among the first resources they go to when any question arises. What is more, these web search engines have for many become the most trusted source of information, more so even than traditional media such as newspapers, news websites or news channels on television. What web search engines present people with thus greatly influences what they believe to be true and consequently it influences their thoughts, opinions, decisions, and the actions they take. It matters a great deal what search engines present people with; more and more our world depends on them [3].

With this in mind, from an information retrieval (IR) research perspective, two things are important. First, it is important to understand what the users are using the

search engines for and secondly, how can we leverage this knowledge about user needs to improve a system's offerings to its users. This thesis is about these two topics: understanding and extracting user tasks and leveraging this task information to enhance user support.

Search behavior, and more generally, information-seeking behavior is often motivated by information needs that prompt search processes that are often lengthy, iterative, intermittent, and characterized by distinct stages, shifting goals and multitasking. One aspect of characterizing information-seekers' goals, contexts and information problems is to consider the *tasks* which have led them to engage in information-seeking behavior, and the tasks that they need to accomplish in information seeking and in information searching, respectively.

The effectiveness of IR systems is measured based on how well users' information problems are resolved and to what extent the information retrieved helps users to achieve their goals. Users' information problems, and their contexts, are various and thus need different types of information support. However, a big disadvantage of current IR systems, including search engines and digital libraries, is that they may not completely comprehend and understand the underlying goal which prompted the user to seek information. As a result, users' interactions with IR systems may be sub-optimal, and returned search results may not help users to achieve their goals. Therefore, it is necessary to explore how IR systems extract and understand user needs and tasks, adapt to a variety of users with different goals, contexts and types of information problems; that is, to contextualize and personalize interaction with IR systems.

The aim of understanding user's goals and tasks pre-dates modern web search, with past research focussing on discovering what people attempt to do in libraries and why, how these activities relate to their more general goals and other characteristics, and their degree of success in their information activities. Belkin *et al.* [4] presented a description of a method for attempting to discover characteristics of the goals, contexts and behaviors of users of libraries, in order to specify the functionality and other characteristics of computerized catalogs which would support them

in their tasks. The problem of user task understanding still remains an important problem in modern era web search.

Contemporary search environments are tailored to support a small set of basic search tasks and provide searchers with few options to search and interact with information, and little to help them synthesize and integrate information across sessions. A major portion of the current generation search systems have been designed to support discrete, transactional searches despite wide recognition that search behavior, and information behavior more generally, is often embedded in and motivated by real world tasks. Given the current state of search systems, accomplishing complex search tasks places intense cognitive burden on the user to explore and discover varied aspects of the task and therefore, (i) explore the domain-space, (ii) identify the necessary sub-tasks involved and (iii) issue queries to accomplish these sub-tasks. Such a multiple stage process becomes prohibitively challenging for searchers who might not necessarily be domain experts and have little to no domain knowledge of the task they're trying to accomplish. As a result, users require significantly more effort and time to complete such tasks [5, 6, 7].

To address the above challenges, the need arises for automated methods for sense-making of search tasks for the development of task based information retrieval systems - search systems which are not only adept at modelling, identifying and extracting tasks but are also capable of supporting user's complex search task missions.

Identifying and representing user tasks properly enables system designers to better understand user interactions and gauge their satisfaction. In light of user's task information, user interaction signals aid in devising search systems that can help end users complete their tasks. Accurate representation of tasks are used to provide users with better query suggestions, offer improved personalization, provide better recommendations, help in satisfaction prediction and search result re-ranking.

In this thesis, we draw inspiration from the importance and numerous possible applications of the task information. In two parts in this thesis, we address two

related themes concerning search task information. In the first part of the thesis, we study methods for understanding and characterizing user's search behavior and develop algorithms for extracting search tasks from log data. In the second part of this thesis, we focus on leveraging the task information in various applications. Specifically, we focus on learning task based user representations, emphasize the use of task context while learning neural embeddings and develop deep sequential algorithms for gauging user's satisfaction with an IR system at the query and task level.

In the next section we outline the research in this thesis and the questions that are answered within it.

1.1 Research Outline & Questions

This thesis focusses on understanding, extracting task information from logged data and leveraging the extracted task information for enhanced representations and satisfaction prediction. As outlined above, we distinguish two research themes on automated methods for sensemaking of search tasks:

In the first part, we begin by considering user interactions with a commercial real world search engine and analyse terascale data comprising of millions of users and search queries. We present insights on user level and task level heterogeneities embedded in search logs and characterize user's multi-tasking behavior. Equipped with an understanding of user interactions and search tasks, we proceed to develop task extraction algorithms. Specifically, we focus on complex tasks and present a bayesian non-parametric approach to extract sub-tasks from a given complex task. Finally, we hypothesize that tasks could recursively be broken down into sub-tasks, and propose a hierarchical bayesian non-parametric approach to extract task-subtask hierarchies.

In the second part of the thesis, we demonstrate how the extracted task information could be leveraged in various applications. We first consider user modelling and personalization and demonstrate that task information is more helpful than traditionally used topical interests in constructing user representations. Beyond user

representations, we consider query representations and present a neural embedding model which leverages task context when learning embeddings. Finally, we move beyond representations and focus on gauging user satisfaction at the query as well as task level. Specifically, we propose a novel deep sequential architecture which makes use of user interaction signals for predicting user's task satisfaction.

Part I: Understanding and Extracting Tasks

The analysis of search behavior over time helps in identifying different queries that express the same underlying information need. Most previous work has focused on search behavior analysis and prediction within a single search session, where a session refers to a sequence of search activities terminated by a prolonged period of inactivity [8]. While existing search engines are adept at handling simple information seeking needs spanning single or a session full of queries, users get little or no help when their information need transcends this boundary. Recent work has started to investigate and analyze multi-session information needs, called search tasks [9, 10, 6, 11, 12]. While past research has relied on varied definition of tasks [12, 11], our definition of search tasks follows from previous work [13] which identified tasks as search missions and goals. More formally,

Definition: A search task is an atomic information need resulting in one or more queries [13].

Cross-session task consists of a series of queries that corresponds to a distinct high-level information need. The queries related to the task are not necessarily consecutive, and a single search session may contain interleaved queries from multiple cross-session tasks, as well as shorter, within-session tasks. Thus, task based search systems help users in tackling more complex informational needs spanning multi-sessions. Often search tasks involve many different but related and necessary aspects which warrant the need of issuing different sets of queries to fulfil those different multi-aspect information needs. It is mostly the case that these independent information needs arise from an overall complex search goal or task a user has,

which prompted the user to attempt tackling these individual information needs. We define such multi-aspect information needs as Complex Search Tasks:

Definition: A complex search task is a multi-aspect or a multi-step information need consisting of a set of related tasks [13, 9].

Complex tasks differ from simple atomic tasks in a number of characteristics including task complexity, time and user effort required, goal difference, cognitive difference, and task areas. Given the wide array of use cases based on which millions of users access IR systems, there is rich inherent diversity not just in the kind of tasks users perform but also in the ways in which different users interact with the system in performing these tasks.

The first part of this thesis is devoted to understanding and extracting search tasks from large scale user interaction data. We begin by user behavior based on large scale user interaction data and extracting search tasks.

Chapter 3. Understanding Search Behavior

We begin our exploration of user behavior and search tasks with a large scale analysis of search log data from a real world commercial search engine. While a major share of prior work have considered search sessions as the focal unit of analysis for seeking behavioral insights, we instead focus on search tasks as our unit of analysis and quantify user search task behavior for both single- as well as multi-task search sessions and relate it to tasks and topics. Multi-tasking within a single online search sessions is an increasingly popular phenomenon. We aim at quantifying, first, the prevalence of multi-tasking behavior in online search sessions (i.e. how common is multi-tasking?), and second, the extent of multi-tasking behavior in multi-task sessions (i.e. how many tasks on average are there in multi-task search sessions?). We also seek to uncover the presence of user-level idiosyncrasies in multi-tasking behavior in search sessions. Specifically, we attempt to understand the proportion of sessions per user that are single tasked vs. multi-tasked. Consequently, we seek to uncover any underlying categorizations among the users based on the extent of

their multi-tasking behavior, i.e., Can we identify and classify groups of users who demonstrate similar proportions of multi-tasking behavior?. The presence of competing or interfering tasks within a single session could accentuate or attenuate the search effort expended by the users. To this end, we wish to understand the relationship between task multiplicity and total effort expended by the users, i.e., do users who multitask more(less) expend more effort than users who multitask less(more)? Going beyond multi-tasking, we characterize the relationship between topics and search tasks. We investigate user-disposition, topic and user-interest level heterogeneities that are prevalent in search task behavior. The insights developed from understanding user's task behavior provides firm grounds for rest of the work described in the thesis.

Chapter 4. Exploiting Distributional Semantics with Non-parametric Priors for Extracting Sub-Tasks

While most prior research in the area of task extraction has focused on segmenting chronologically ordered search queries into higher level search tasks, a more naturalistic viewpoint involves viewing query logs as convoluted structures of tasks-subtasks, with complex search tasks being decomposed into more focused subtasks. This chapter focusses on complex search tasks and investigates the potential of breaking down a complex task into simpler sub-tasks. Specifically, we address the problem of extracting sub-tasks from a given collection of on-task search queries. Subtask identification turns out to be a complex problem due to multiple reasons, including unknown number of subtasks and the strong overlap in the informational needs embodied by the different subtasks. A novel generative model based on coherence estimates is proposed to identify and extract semantically cohesive subtasks.

Chapter 5. Extracting Hierarchies of Tasks-Subtasks

Task extraction is quite a challenging problem as search engines can be used to achieve very different tasks, and each task can be defined at different levels of granularity. A major limitation in existing task-extraction methods lies in their treatment

of search tasks as flat structure-less clusters which inherently lack insights about the presence or demarcation of subtasks associated with individual search tasks. In reality, often search tasks tend to be hierarchical in nature. For example, a search task like planning a wedding involves subtasks like searching for dresses, browsing different hairstyles, looking for invitation card templates, finding planners, among others. Each of these subtasks (1) could themselves be composed of multiple subtasks, and (2) would warrant issuing different queries by users to accomplish them. Hence, in order to obtain more accurate representations of tasks, new methodologies for constructing hierarchies of tasks are needed. This chapter considers the challenge of extracting hierarchies of search tasks and their associated subtasks from a search log given just the log data without the need of any manual annotation of any sort. We present an efficient Bayesian nonparametric model for discovering hierarchies and propose a tree based nonparametric model to discover this rich hierarchical structure of tasks/subtasks embedded in search logs.

Part II: Leveraging Task Information

While user behaviours are largely determined by their own goals, tasks and preferences, the mined knowledge about user tasks and information needs from log activity data reveals different user intentions and behaviour patterns, which provide unique signals for user centric optimization and personalization. In Part II of this thesis, we investigate whether and how the extracted task information be leveraged to improve search systems.

We begin by focussing on the goal of learning user models for personalization, and address the question - how can user's task information be used to develop better user models and representations? Second, we investigate the benefits of search context in learning query representations and ask the research question whether task information provides better context for IR systems to learn from. Finally, moving beyond user modelling and query representations, we consider the problem of user satisfaction prediction and investigate how useful user interaction signals are in predicting searcher's task satisfaction.

Chapter 6. Task based User Modelling

Given the distinct preferences of different users while using search engines, search personalization has become an important problem in information retrieval. Most approaches to search personalization are based on identifying topics a user may be interested in and personalizing search results based on this information. While topical interests information of users can be highly valuable in personalizing search results and improving user experience, it ignores the fact that two different users that have similar topical interests may still be interested in achieving very different tasks with respect to this topic (e.g. the type of tasks a broker is likely to perform related to finance is likely to be very different than that of a regular investor). Hence, considering user's topical interests jointly with the type of tasks they are likely to be interested in could result in better personalised experience for users.

In this chapter, we postulate that in a web search setting, a user representation based on the search tasks users' perform as well as their topical interests would better capture user actions, interests and preferences. While topical interests capture the heterogeneity among users stemming from varied topical interests, such task based approaches would assist in capturing the heterogeneity stemming from differences in user needs & behaviors. We present an approach that uses search task information embedded in search logs to represent users by their actions over a task-space as well as over their topical-interest space. In particular, we describe a tensor based approach that represents each user in terms of (i) user's topical interests and (ii) user's search task behaviours in a coupled fashion and use these representations for personalization. Additionally, we also integrate user's historic search behavior in a coupled matrix-tensor factorization framework to learn user representations.

Chapter 7. Task based Embeddings

Continuous space word embedding have been shown to be highly effective in many information retrieval tasks. Embedding representation models make use of local information available in immediately surrounding words to project nearby context words closer in the embedding space. With rising multi-tasking nature of web search sessions, users often try to accomplish different tasks in a single search ses-

sion. Consequently, the search context gets polluted with queries from different unrelated tasks which renders the context heterogeneous. In this work, we investigate the research question whether task information provides better context for IR systems to learn from. A novel task context embedding architecture is proposed to learn representation of queries in low-dimensional space by leveraging their task context information from historical search logs using neural embedding models.

Chapter 8. Task based Satisfaction

Detecting and understanding implicit signals of user satisfaction are essential for experimentation aimed at predicting searcher satisfaction. Search tasks help us not only to capture searcher's goals but also in understanding how well a system is able to help the user achieve that goal. However, a major portion of existing work on modelling searcher satisfaction has focused on query level satisfaction. The few existing approaches for task satisfaction prediction have narrowly focused on simple tasks aimed at solving atomic information needs. This chapter goes beyond such atomic tasks and consider the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks. Specifically, we investigate the research question - how can we best leverage user interaction signals to gauge user's task satisfaction?

We begin by considering holistic view of user interactions with the search engine result page (SERP) and extract detailed interaction sequences of their activity. We then look at query level abstraction and propose a novel deep sequential architecture which leverages the extracted interaction sequences to predict query level satisfaction. Further, we enrich this model with auxiliary features which have been traditionally used for satisfaction prediction and propose a unified multi-view model which combines the benefit of user interaction sequences with auxiliary features.

Finally, we go beyond query level abstraction and consider query sequences issued by the user in order to complete a complex task, to make task level satisfaction predictions. A number of functional composition techniques are proposed which take into account query level satisfaction estimates along with the query sequence to predict task level satisfaction. We investigate how good the proposed deep se-

quential models are and how they compare with existing state-of-the-art implicit signal based satisfaction prediction models.

1.2 Main Contributions

In this section we summarize the main contributions of this thesis. Our contributions come in the form of analysis, algorithmic and empirical contributions.

1.2.1 Analysis & Algorithmic Contributions

We list five algorithmic contributions and one large scale analysis. The analysis is from user interaction data from a real world search engine. The first two algorithmic contributions are task extraction techniques that focus on complex tasks and sub-tasks. The third algorithmic contribution preents a tensor based approach for learning task based user representations. Algorithmic contributions 4 and 5 present deep learning based approach for learning neural query embeddings and for predicting task satisfaction, respectively.

1. **A large scale study of search tasks.** In Chapter 3 we investigate a large scale search log data from a major US based search engine for a period of one month spanning a sample of over 2 million users, and 200 million search sessions. We present insights on user's search behaviour and their multi-tasking habits. Furthermore, we present insights on how tasks are related to user level and topic level heterogeneities present in user's search behavior.

2. **Unsupervised generative model for extracting sub-tasks.** In Chapter 4 we propose a novel bayesian non-parametric generative model based on Chinese Restaurant Processes to extract sub-tasks from a complex search task. The proposed method is able to automatically identify the appropriate number of sub-tasks and extract coherent subtasks.

3. **Bayesian non-parametric method for task hierarchies.** In Chapter 5 we present a novel bayesian non-parametric hierarchical algorithm to extract recursive hierarchies of search tasks and subtasks. The proposed model is able

to leverage insights present in the search log to data to extract rich non-binary arbitrary shaped task hierarchies in an unsupervised fashion.

4. **Coupled matrix-tensor model for personalization.** In Chapter 6 we propose a model that combines topic based user modelling with task based user models and propose a coupled matrix-tensor factorization model which jointly learns user representations based on user's search history, term usage behavior, topical interest profiles and search task behaviors.

5. **Neural embedding model for query representations.** In Chapter 7 we propose a novel task based embedding architecture to learn distributed semantic representation of query terms which prefers task context over local information in immediately surrounding words. The proposed model demonstrates that embeddings learned on a task-constrained context perform better than the traditionally used global or session context.

6. **Deep sequential architecture for Task satisfaction.** In chapter 8 we propose a novel deep sequential architecture which leverages user's detailed interaction sequences and enriches this information with auxiliary interaction features into a unified multi-view model for query satisfaction prediction. Furthermore, we go beyond query level abstraction and consider query sequences issued by the user in order to complete a complex task, to make task level satisfaction predictions. We propose a number of functional composition techniques which take into account query level satisfaction estimates along with the query sequence to predict task level satisfaction.

1.2.2 Empirical Contributions

The research presented in this thesis focusses on developing advanced models of search tasks and encapsulating user's task information into different applications. We list a total of eight empirical contributions resulting from this research. The first empirical contribution investigates user's search behavior, characterizes user groups and quantifies the extent of multi-tasking in web search. Empirical contri-

butions 2 and 3 compare the proposed subtask extraction and task hierarchy extraction methods with established baselines. Contributions 5 through 8 discuss various applications of the extracted tasks, in a number of ways including use modelling (Contribution 5), query representations (Contribution 6), and user satisfaction prediction (Contribution 7 and 8).

1. **Multi-tasking Behavior of Users.** In Chapter 3, we derive insights based on real world search logs and establish user types based on their search behaviors and establish the need for considering tasks as the atomic unit of investigation. We quantify the extent of multi-tasking prevalent in web search and characterizing the multi-tasking behavior of users. Additionally, we identify user level and topic level heterogeneity and present three user groups based on their multi-tasking habits: (i) Focussed, (ii) Multi-taskers, and (iii) Super-taskers. Finally, we show how users exercise specific multi-tasking preferences when searching for topics that are of high vs. low interest to them.

2. **Extracting Coherent Sub-Tasks.** In Chapter 4, we empirically demonstrate that the bayesian non-parametric approach for finding subtasks extracts subtasks which are pure and coherent. Further, a user judgment study is conducted which establishes that the subtasks re indeed valid and help solve related information needs. Finally, based on a large scale log analysis of over 2 million users, we demonstrate that users expend different efforts in accomplishing the different subtasks, and that there significant differences in task effort metrics across the sub-tasks.

3. **Evaluating Task Hierarchies.** In Chapter 5, we present a number of evaluation strategies to evaluate the quality, validity and usefulness of the task hierarchies extracted. Based on a labelled task dataset, we demonstrate that the proposed hierarchical task extraction method achieves competitive performance and is able to extract hierarchies of subtasks wherein the subtasks are coherent, valid and useful.

4. **Utility of Task Hierarchy.** In Chapter 5, we present experiments on query

term prediction using TREC Session Track and AOL datasets. Given the initial queries from a user session and a set of tasks extracted from Session Track data, we leverage queries from the identified task to predict future query terms.

5. **Learning User Representations.** In Chapter 6, we demonstrate that it is possible to represent users in a joint topic-task-term space via coupled tensor-matrix model, and demonstrated that coupling user's task information with their topical interests indeed helps us build better user models. Further, task based user representations helps in identifying better user cohorts and demonstrate that user clusters obtained from via using topic-task coupled representations indeed perform better than the clusters obtained via just Bag-of-Terms or task baselines.

6. **Task based Query Suggestions.** In Chapter 7, we present results on task based query suggestions wherein query representations are learnt based on a neural embedding model which leverages task context. Empirical results based on TREC Tasks Track data from 2015 and 2016 demonstrates that task based query representations indeed help in suggestion more task-relevant suggestions.

7. **User Interaction Sequences.** In Chapter 8, we consider detailed user interaction sequences and show that the proposed deep sequential model based on user interaction sequences is better at predicting user satisfaction at the query level than click based and other static mouse gesture based signals. We also demonstrate that jointly modelling interaction sequence with static interaction signals is better than only considering interaction sequences.

8. **Task Satisfaction Prediction.** In Chapter 8 we go beyond query level abstraction and consider query sequences to make task level satisfaction predictions. We evaluate how different functional compositions approaches perform when predicting task level satisfaction from individual query satisfaction estimates. First we show that the multi-view deep sequential model is better able

to predict task satisfaction than a number of neural and non-neural baselines, including CRFs, generative models and simple LSTM architectures. Second, while evaluating the performance of functional composition techniques, we show that the most lenient aggregating technique (Maximum) consistently achieves higher accuracy than the most strict satisfaction criterion (Minimum) and that the differential weighting scheme performs better than the average function, which hints at the fact that not all queries contribute the same towards a task.

1.2.3 Community Contributions

Beyond the analysis, algorithmic and empirical contributions, we also enlist few contributions made to the overall research community as part of the research done in this thesis.

1. TREC Tasks Tracks: Based on the research on tasks, a new TREC track was proposed and run for three years (2015-2017): TREC Tasks Track. The primary goals of the track were to evaluate system's understanding of tasks users aim to achieve and evaluate relevance of retrieved documents with respect to underlying tasks in query. Datasets containing queries with tagged tasks were released as part of the process. Overview summaries describe the categories of evaluation mechanisms used in the track along with the corpus, topics, and tasks that comprise the test collections [14, 15].

2. CIKM 2017 Tutorial: A tutorial on *Understanding Inferring User Tasks and Needs*[1] describing the state-of-the-art techniques in understanding and extracting tasks was proposed and will be presented at CIKM 2017.

3. WSDM 2018 Workshop: A workshop on *Learning from User Interactions*[2] is being organized to provide a forum for academic and industrial researchers working at the intersection of user understanding, search tasks and user interactions to discuss the research challenges and directions of future research.

[1] https://task-ir.github.io/Task-based-Search/
[2] https://task-ir.github.io/wsdm2018-learnIR-workshop/

1.3 Thesis Overview

This section provides an overview of this thesis. We finish this section with reading directions.

The first chapter, to which this section belongs, gives an introduction to the subject of this thesis. This chapter also provides an overview of the research questions, the contributions and the origins of this work. Chapter 2 then introduces the background and related work for all six research chapters that follow. The core of this thesis consists of two parts.

In Part I of this thesis, we study user's search behavior and present algorithms for extracting search tasks from logged search interaction data. An overview of user interaction with search systems is provided in Chapter 3. It build upon search sessions to identify and characterize search tasks and topics. In characterizing these search tasks across sessions, it considers the different distinct forms of heterogeneity inherent in the search-task behavior and additionally provide a detailed analysis of multi-tasking behavior for different user groups, and across multiple session sessions. Query logs can be viewed as convoluted structures of tasks-subtasks with complex search tasks being decomposed into more focused sub-tasks. In chapter 4, we focus on extracting sub-tasks from a given collection of on-task search queries. Complex tasks often tend to have multiple subtasks associated with them and a more naturalistic viewpoint would involve viewing query logs as hierarchies of tasks with complex search tasks being decomposed into more focused sub-tasks. In chapter 5, we propose an efficient Bayesian nonparametric model for extracting hierarchies of such tasks and subtasks.

Part II of this thesis revolves around leveraging the extracted task information in different applications. Considering user's topical interests jointly with the type of tasks they are likely to be interested in could result in better personalised experience for users. In chapter 6, we present a coupled tensor-matrix factorization approach that uses search task information embedded in search logs to represent users by their actions over a task-space as well as over their topical-interest space. Chapter 7 discusses a novel task based embedding model which leverages task context to

learn query representations. Chapter 8 focusses on the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks. It proposes a multi-view deep sequential model for query as well as task satisfaction prediction.

Lastly, Chapter 9 concludes this thesis, where we summarize the content and findings of this thesis, discuss the limitations of the presented work, and briefly reflect onto future work.

The two parts of this thesis (Part I and Part II) are self-contained and form independent parts. Readers familiar with the background on web search and information retrieval can skip the corresponding sections of Chapter 2 and glance through the machine learning approaches briefed in Chapter 2 to develop a better understanding of the algorithmic background needed for understanding the proposed algorithms. Part II assumes basic understanding of search tasks and in particular, basic understanding of any task extraction technique.

1.4 Origins

We list for each research chapter the publications on which it is based. For each publication we mention the role of each co-author. The thesis is based on in total 8 publications. In addition, it draws on ideas from six others.

1. The first part of Chapter 3 is based on *Characterizing Users' Multi-Tasking Behavior in Web Search* [16], published at CHIIR 2016 by Mehrotra, Bhattacharya and Yilmaz. Mehrotra implemented the analysis components and performed the experiments. All authors contributed to the text.

2. The second part of Chapter 3 is based on *Sessions, Tasks & Topics - Uncovering Behavioral Heterogeneities in Online Search Behavior* [17], published at SIGIR 2017 by Mehrotra, Bhattacharya and Yilmaz. Mehrotra performed most of the experiments and analysis while all authors contributed to the text.

3. Chapter 4 is based on *Deconstructing Complex Search Tasks: a Bayesian Nonparametric Approach for Extracting Sub-tasks* [18] published at NAACL

2016 by Mehrotra, Bhattacharya and Yilmaz. Part of the chapter are also based on an extension of the work, *Exploiting Distributional Representations with Distance Dependent CRPs for Extracting Sub-Tasks*, which is currently under review. Mehrotra implemented the algorithm and performed experiments. All authors contributed to the text.

4. Chapter 5 is based on *Extracting Hierarchies of Search Tasks & Subtasks via a Bayesian Nonparametric Approach* [19] published at SIGIR 2017 by Mehrotra and Yilmaz; . Preliminary work of this research originally appeared in, "Towards hierarchies of search tasks & subtasks" [20], which was published as a Poster at WWW 2015. Mehrotra performed the experiments and implemented the algorithms. All authors contributed to the text.

5. Chapter 6 is based on *Terms, Topics & Tasks: Enhanced User Modelling for Better Personalization* [21], published at ICTIR 2015 by Mehrotra and Yilmaz. All authors contributed to the text, while Mehrotra implemented the model and performed the experiments.

6. Chapter 7 is based on *Task Embeddings: Learning Query Embeddings using Task Context* [22], published at CIKM 2017 by Mehrotra and Yilmaz. Mehrotra implemented the model while both authors contributed to the text.

7. Chapter 8 is based on *Deep Sequential Models for Task Satisfaction Prediction* [23], published at CIKM 2017 by Mehrotra, Awadallah, Shokouhi, Yilmaz, Zitouni, Kholy and Khabsa. Mehrotra implemented the code for the deep sequential models, and ran the experiments. Kholy, Khabsa and Mehrotra contributed to collecting judgment labels by crowdsourced study, and all authors contributed to the text.

This thesis also, but indirectly, builds on publications on *extracting interaction subsequences* [24], *exploring digital assistants tasks* [25], *auditing search engines for fairness* [26], *identifying user sessions in digital assistants*, *predicting supply side engagement* [27], *TREC Tasks Track* [14, 15] and *query selection for learning to rank* [28].

Other work, not directly related to this thesis, did contribute to insight in the broader research areas of *causal dependencies in information seeking* [29], *topic models* [30], *learning sparse representations* [31] and *valence prediction in news* [32].

Chapter 2

Background

The work presented in this thesis sits at the crossing between the fields Information Retrieval (IR) and Machine Learning (ML). This thesis deals with understanding search tasks, enabling and improving task extraction and leveraging task information for enhanced user modelling, interaction understanding and satisfaction prediction. In this chapter, we present all relevant background and related work that serves as a basis for understanding this thesis and sets stage for the research chapters in this thesis. In Section 2.1, we start with a historical introduction into the field of information retrieval (IR) including description of modern IR methods and systems. We then continue with understanding user sessions and tasks in Section 2.2, user representations and satisfaction in Section 2.3, and finally algorithmic background on bayesian non-parametrics, distributed representations and hierarchical models in Section 2.4. Section 2.1 serves as background to the whole thesis. Sections 2.2 provides background to Part I and section 2.3 serve as background to Part II. Section 2.4 provides background context of some of the algorithmic approaches employed in this thesis.

2.1 Information Retrieval

For thousands of years people have realized the importance of archiving and finding information. With the advent of computers, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. The field of Information Retrieval (IR) was born in the 1950s out

of this necessity. Over the last seventy years, the field has matured considerably. Several IR systems are used on an everyday basis by a wide variety of users. This section provides a brief overview of the key advances in the field of Information Retrieval, and a description of modern day information retrieval systems.

2.1.1 Brief History

Conventional approaches to managing large collections of information originate from the discipline of librarianship. Commonly, items such as books or papers were indexed using cataloguing schemes. Eliot and Rose claim this approach to be millennia old: declaring Callimachus, a 3rd century BC Greek poet as the first person known to create a library catalogue [33]. Facilitating faster search of these physical records was long researched, for example, Rudolph filed a US patent in 1891 for a machine composed of catalogue cards linked together, which could be wound past a viewing window enabling rapid manual scanning of the catalogue. Soper filed a patent for a device in 1918 [34], where catalogue cards with holes, related to categories, were aligned in front of each other to determine if there were entries in a collection with a particular combination of categories. If light could be seen through the arrangement of cards, a match was found.

The idea of using computers to search for relevant pieces of information was popularized in the article As We May Think by Vannevar Bush [35] in 1945. It would appear that Bush was inspired by patents for a *statistical machine*. In the 1920s the German scientist Emanuel Goldberg pioneered the electronic retrieval technology and library automation. Goldberg designed, built and demonstrated a *photoelectric microfilm selector* which contained many, if not all, of the concepts history of science professionals now associate with Vannevar Bush. It seems this was the first practical application of electronics to the selection of data on film. In 1914, Emanuel Goldberg developed a machine that read characters and converted them into standard telegraph code (early OCR). Later (by May 1927) Goldberg designed a photoelectric microfilm selector, which he called a *statistical machine*. Two prototypes were built at Zeiss Ikon by 1931 and, perhaps, constitute the first successful electronic document retrieval.

Mooers [36] was the first to introduce the term and problem of "*information retrieval*" in the 1950s when he introduced it to this scientific community. Mooers not only introduced the problem but he also had ideas about how a retrieval systems should be evaluated based on whether the systems do their job; and how expensive is it to operate such a system.

The first description of a computer searching for information was described by Holmstrom at a specially convened conference was held by the UKs Royal Society in 1948. He described a "machine called the Univac" capable of searching for text references associated with a subject code. The code and text were stored on a magnetic steel tape [37]. Holmstrom stated that the machine could process "at the rate of 120 words per minute". Automated information retrieval systems were introduced in the 1950s: one even featured in the 1957 romantic comedy, Desk Set, which drew public attention to the innovation and was centred on a group of reference librarians who were about to be replaced by a computer. IR as a research discipline was starting to emerge at this time with two important developments: how to index documents and how to retrieve them. In the 1960s, the first large information retrieval research group was formed by Gerard Salton at Cornell. By the 1970s several different retrieval techniques had been shown to perform well on small text corpora such as the Cranfield collection (several thousand documents) [38]. Large-scale retrieval systems, such as the Lockheed Dialog system, came into use early in the 1970s.

The 1970s and 1980s saw many developments built on the advances of the 1960s. Various models for doing document retrieval were developed and advances were made along all dimensions of the retrieval process. These new models/techniques were experimentally proven to be effective on small text collections (several thousand articles) available to researchers at the time. However, due to lack of availability of large text collections, the question whether these models and techniques would scale to larger corpora remained unanswered. This changed in 1992 with the inception of Text Retrieval Conference, or TREC [39]. TREC is a series of evaluation conferences sponsored by various US Government agencies under the auspices

of NIST, which aims at encouraging research in IR from large text collections. The algorithms developed in IR were the first ones to be employed for searching the World Wide Web from 1996 to 1998.

2.1.2 Modern Day Information Retrieval

To the surprise of many, the field of information retrieval has moved from being a primarily academic discipline to being the basis underlying most people's preferred means of information access. Relentless optimization of information retrieval effectiveness has driven web search engines to new quality levels where most people are satisfied most of the time, and web search has become a standard and often preferred source of information finding.

In response to various challenges of providing information access, the field of information retrieval evolved to give principled approaches to searching various forms of content. The field began with scientific publications and library records, but soon spread to other forms of content, particularly those of information professionals, such as journalists, lawyers, and doctors. Much of the scientific research on information retrieval has occurred in these contexts, and much of the continued practice of information retrieval deals with providing access to unstructured information in various domains.

Modern IR as we know it started at the 1958 International Conference on Scientific Information, as recalled by Sparck Jones [40]. Following that conference, researchers considered automating retrieval tasks that were manual tasks until then [41]. This progress combined with the growing amounts of scientific literature culminated in the Cranfield experiments by Cleverdon [42], setting the stage for much IR research today.

Prior to web search engines, catalogs have been the primary device by which people gain access to content of libraries. For most of the twentieth century, this device has been in the for most of the familiar card catalog. The advent of library automation systems in the 1960s and 1970s, originally for such housekeeping tasks as acquisition and circulation control, provided the opportunity for a new form of library catalog, which is now known generically as the Online Public Access Catalog

(OPAC).

Since Mooers coined the term "*Information Retrieval*", the field of IR rapidly evolved, driven by the need to search through the ever larger volumes of information that are being produced and stored. This was further accelerated with the advent of the internet and the increased access to digital equipment - such as personal computers and recently mobile devices - by the masses. Despite the intensified research in recent years, many of the methods developed in the early years of IR research are still very central to modern IR systems. We therefore describe them in some detail here, starting with a widely accepted definition of IR, which we borrow from Manning et al. [43]: "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

A searcher approaches an IR system with a need for information derived from an "*anomalous state of knowledge*". So, in essence, IR starts with an information need. An information need [44] is born in the head of a person and can be unrecognized by the person as such. If this person would use an IR system to satisfy the information need, we would refer to this person as a user. The act of retrieving information used to be a librarian's task. A user would explain the information need to a librarian who would then know how to search through a collection of books and articles. Nowadays the more common scenario is that a user translates their need into keywords and enters them into a search engine.

While efforts into understanding user needs and tasks have been around for few decades [4], the problem remains challenging to this day. Indeed, as search systems have evolved, so have user's expectations of the kind of information and the ease with which they can find that information. Consequently, information retrieval systems are now able to help searchers on a wide array of information needs, from simple weather queries to complex conversational queries and interactions. With internet going mainstream, increasingly larger proportions of users are now using search engines to solve their information needs. Over the years, domain specific search portals have been developed which focus on a particular type of content

and answer domain specific search queries. Further, the traditional web search have moved beyond simple textual queries to incorporate richer content including images, media, sound, locations among others. The results shown to the users have also advanced to include richer content like entity panels, custom answers and different verticals.

Recent years have witnessed a gradual shift towards searching and presenting the information in a conversational form. Chatbots, personal assistants in our phones and eyes-free devices are being used increasingly more for different purposes, including information retrieval and exploration. With improved speech recognition and information retrieval systems, more and more users are increasingly relying on such digital assistants to fulfil their information needs and complete their tasks.

The notion of *tasks* helps in providing an interpretable abstraction for grounding user interactions with not just traditional search environments but also with novel interaction interfaces. Given the importance of understanding user needs and tasks, this thesis focusses on extracting task information and leveraging it to develop better search systems. In this thesis, we consider the problem of understanding user's task behaviors and present algorithms to extract search tasks and task-subtask hierarchies. Additionally, we go beyond task extraction and present novel applications of the task information in a number of applications, including user modelling, query representations and user satisfaction prediction. In the next two sections, we briefly overview the background material required to understand and appreciate the contributions made in this thesis. We begin by a providing gentle background on log data analysis, search sessions, context and tasks. We then proceed to consider the user centric view, and discuss prior work on user representations, interpreting user interaction signals and user satisfaction.

2.2 Understanding User Interaction Logs

Web search logs provide explicit cues about the information seeking behavior of users. As a result, web logs have been extensively studied to generate insights that

would improve the search experiences of users. We next describe prior work on analyzing web search logs to extract search sessions, understand search tasks, characterize user's multitasking behavior and provide support and assistance to users engaged in satisfying complex information needs.

Information retrieval (systems target helping people find information. However, in everyday life, searching for information is often driven by motivating goals, such as to accomplish some work task at hand [45, 46].

2.2.1 Search Sessions

Session identification is a common strategy used to develop metrics for web analytics and perform behavioral analyses of user-facing systems. Sessions allow us to look beyond individual queries, preserve semantic associations between query trails and maintain context of user activity. Strategies for session identification from log data have been extensively studied. Content based heuristics [47] exploit lexical content of queries for determining topical shift in query streams. Navigation-oriented heuristics [48] involve inferring browsing patterns based on the HTTP referrers and URLs associated with each request by a user. Time-oriented heuristics [49] refer to the assignment of an inactivity threshold between logged activities to serve as a session delimiter. The assumption implied is that if there is a break between a user's actions that is sufficiently long, it's likely that the user is no longer active, the session is assumed to have ended, and a new session is created when the next action is performed.

Catledge & Pitkow [50] were among the first to use client-side tracking to examine browsing behavior and propose time based threshold. They reported that the mean time between logged events 9.3 minutes and chose to add 1.5 standard deviations to that mean to achieve a 25.5 minutes inactivity threshold. Over time this threshold has simplified to 30 minutes. This is the most popularly-used approach to identify sessions, with 30 minutes serving as the most common threshold [51, 48, 52]. In addition, Radlinski and Joachims [53] used a 30-minute timeout together with query similarity measures to define sequences of similar queries that combine to form so-called query chains.

While session identification helps in understanding user behavior at some level, they offer limited support to help understand the overall task which prompted user interaction at the first place. In order to understand the inherent search task, methods relying on just time-oriented heuristics are of limited utility. In a later section, we discuss recent effort aimed at extracting and identifying search tasks.

2.2.2 Search Context

There is a growing body of work examining how knowledge of a searcher's interests and search context can be used to improve various aspects of search. The information retrieval (IR) community has theorized about context [54], developed context-sensitive search models [55], leveraged context to predict user interests [8] and performed user studies investigating the role of context in the search process [56]. Using the context of user activities within a search session has also been used to improve query analysis. Short queries are often ambiguous, so researchers have used previous queries and clicks in the same session to build a richer models of interests and improve how the search system interprets users' information needs. Cao et al. [57, 58] represented search context by modeling sessions as sequences of user queries and clicks. They learned sequential prediction models such as hidden Markov models and conditional random fields from large-scale log data, and applied the models to query suggestion, query categorization, and URL recommendation. Mihalkova and Mooney [59] used similar search session features to disambiguate the current query.

2.2.3 Search Tasks

The effectiveness of information retrieval (IR) systems is measured based on how well users' information problems are resolved and to what extent the information retrieved helps users to achieve their goals. One aspect of characterizing information-seekers' goals, contexts and information problems is to consider the tasks which have led them to engage in information-seeking behavior, and the tasks that they need to accomplish in information seeking and in information searching, respectively.

The aim of understanding user's goals and tasks pre-dates modern web search, with past research focussing on discovering what people attempt to do in libraries and why, how these activities relate to their more general goals and other characteristics, and their degree of success in their information activities. Belkin *et al.* [4] presented a description of a method for attempting to discover characteristics of the goals, contexts and behaviors of users of libraries, in order to specify the functionality and other characteristics of computerized catalogs which would support them in their tasks. The problem of user task understanding still remains an important problem in modern era web search.

There have been many studies on task type classification [60, 61, 46], and Li & Belkin [46] have proposed an extensive scheme to classify tasks based on many dimensions of task features. Among the many task features, for work tasks which consist of multiple sub-tasks, the relationship between the subtasks seems salient and it is necessary to take it into account because the orders of sub-tasks may vary during the process of accomplishing the work tasks.

Tasks that drive people to engage in information seeking are not restricted to those which are strictly work-related, but include various sorts of nonwork information-seeking activities in individuals' everyday lives. For instance, everyday life information seeking (ELIS) has been attracting increasing research attention. Previous studies in this area have investigated aspects of seeking orienting information from media [62], planning for a vacation trip [63], and others like shopping, weather, transportation, etc. [64]. Such a phenomenologically informed approach provides novel ideas for IR research. It helps clarify the preference and relevance criteria for information seeking by extending the evaluation base from the narrower search task to the broader context of people's everyday lives, which may be more suitable in the situation of interactive IR [65].

Researchers have spent a fairly extensive amount of effort examining the effects of different tasks on information searchers' behaviors and performance. A common approach is to classify user tasks into different types along some task feature(s). These include, for example: closed versus openended tasks [66]; specific

versus general tasks [67]; factual, descriptive, instrumental, and exploratory tasks [68]; fact-finding versus information gathering [69]; and learning about a topic, making a decision, finding out how to, finding facts, and finding a solution [60]. The various standards and definitions of task classification make it difficult to compare findings across studies. This makes it necessary to have some standard classification schemes. A rather comprehensive classification scheme is provided by Li and Belkin [46], which includes a number of dimensions: task product, objective complexity, subjective complexity, and difficulty, to name a few.

There has been a large body of work focused on the problem of segmenting and organizing query logs into semantically coherent structures. There have been attempts to extract in-session tasks [13, 70, 71], and cross-session tasks [10, 6] from query sequences based on classification and clustering methods.

Lucchese *et al.* [72, 70, 73] exploit the collaborative knowledge collected by Wiktionary and Wikipedia for detecting query pairs that are not similar from a lexical content point of view, but actually semantically related and propose several variants of well known clustering algorithms, as well as a novel efficient heuristic algorithm for extracting tasks from a given query collection. Kotov et al [6] and Agichtein et al [74] studied the problem of cross-session task extraction via binary same-task classification, and found that different types of tasks demonstrate different life spans. Unfortunately, pairwise predictions alone cannot generate the partition of tasks, and post-processing is needed to obtain the final task partitions [75]. Finally, authors in [76] model query temporal patterns using a special class of point process called Hawkes processes, and combine topic model with Hawkes processes for simultaneously identifying and labelling search tasks. Tolomei *et al.* [77] investigated the concept of taskflows and analyzed a large scale query log to generate task based query suggestions.

Jones et al. [13] was the first work to consider the fact that there may be multiple subtasks associated with a user's information need and that these subtasks could be interleaved across different sessions. However, their method only focuses on the queries submitted by a single user and attempts to segment them based on whether

they fall under the same information need. The proposed approach considers solving the task boundary identification and same task identification problem and finds limited applicability in being used for task extraction.

Recent studies suggest that users searches may have multiple goals or topics and occur within the broader context of their information-seeking behaviors [78]. Through an online survey, Wang *et al.* [79] show that 92% of the participants had online sessions where they accessed several sites, to perform between 2 to 8 tasks. In the context of web search sessions, most work on multi-tasking has been based on user studies [70][80]. Other works do not explicitly refer to online multitasking, but provide useful insights. For instance, users access different sites during a session [81] and a large proportion of pages are visited more than once. In addition, the frequency at which a page is revisited differs depending on user habits and the type of website [81][82], or in other words, the web tasks a user accomplishes on the site. More recent studies suggest that users often seek to complete multiple search tasks within a single search session [70] with over 50% of search sessions having more than 2 tasks. At the same time, certain tasks require significantly more effort, time and sessions to complete with almost 60% of complex information gathering tasks continued across sessions [74, 83].

2.2.4 Supporting Complex Search Tasks

Some previous attempts have been made to support people engaged in complex tasks by allowing them to take notes and record results that they already examined [5], or to provide task continuation assistance, whereby the search engine can predict that a searcher is likely to resume a task and hence preemptively save and retrieve the current search state on the searcher's behalf [84]. While these are good ways to support long term tasks, they do not help searchers directly explore or identify potential next steps for their tasks. Other research efforts have focused on building tours or trails to guide the searcher through their search process [85]. While useful, the methods proposed to date have involved restricted domains or hypertext corpora rather than Web search, or have retrieved focused trails of URLs rather than lists of search results[86]. Prior work also studied the problem of predicting the

next search action based on the current actions, either by predicting the next result click[87] or by predicting searchers' short-term interests at a more general level of abstraction (e.g., topical categories[8].

Baraglia *et al.* [88] introduced the notion of search shortcuts and offered query suggestions to drive users towards their goals. Lucchese *et al.* [89] studied the concept of related tasks, and introduced the Task Relation Graph as a representation of users' search behaviors on a task-by-task perspective. The task relation graph is used to construct a task recommender system, which suggests related tasks to users on the basis of the task predictions derived from the task relation graph.

The quality of most of the supporting mechanisms depend on forming accurate representation of tasks, which is the problem being addressed in Part I of this thesis. While most research has focused on extracting task clusters from query logs, this thesis goes a step beyond and considers the problem of complex search and presents algorithms to decompose complex tasks into simpler sub-tasks.

So far, we have discussed the notions of search sessions, context and tasks which are useful in understanding user behavior from search interaction logs. An alternative view of search log data is composed of the user centric view, which makes use of the concepts defined so far (sessions, context, tasks) in building models and representations of users, capture and interpret their interaction signals, as well as decipher whether or not the users are satisfied with system's performance. In the next section, we briefly discuss background work for each of these.

2.3 Understanding User Signals

The research discussed so far discusses user sessions, context and tasks, which helps us understand and comprehend user's information needs. However, while interacting with a retrieval system, users leave behind traces of their activity which provide us insights about their interests, and helps us develop user models. Beyond gauging interests, researchers have leveraged such activity traces to detect implicit feedback signals to gauge and predict user satisfaction. The algorithms and frameworks developed in this thesis build upon, extends and evaluates against past work in these

areas.

2.3.1 User Representations

Irrespective of where the user's data comes from, a model must encode this data. A variety of such models have been used in the past including a vector of weighted terms (e.g. [90]), a set of concepts (e.g. [91]), using topic models (e.g. [92]) or a hierarchical category tree based on ODP and corresponding keywords (e.g. [93]).

Teevan *et al.* [94] constructed user profiles from indexed desktop documents and showed that this information could be used to re-rank search results and improve relevance for individuals. Matthijs and Radlinski [90] constructed user profiles using users' browsing history, and evaluated their approach using an interleaving methodology. Their approach focused on using term based user profiles which often limit the scope of personalization as different users inherently follow different distributions over words. Dou *et al.* [91] investigated a number of heuristics for creating user profiles and generating personalized rankings. Bennett *et al.* [93] made use of hand picked Open Directory Project (ODP) topical categories to construct user profiles. While such topical categories are easily specified, much human effort is required in labelling queries for each topic. ODP categories based methods restricts topic coverage in a major way as search logs offer much richer content both in terms of the number of topics involved as well as the granularity level of each topic. Very recently, Wang *et al* [95] have proposed a generative model which models users as a mixture over latent user groups wherein each group shares a common distribution over queries and a common click preference pattern. Finally, Harvey *et al.* [92] use the topic model based approach to build user profiles from topics obtained and personalize search results based on the learnt user profiles.

Aiming for short-term personalization, Sriram et al. [96] describe a search engine that personalized based on the current user session. A longer term personalization click model can also be used, exploting clickthrough data collected over a long time period. For example, Speretta and Gauch [97] and Qiu and Cho [98] model users by classifying previously visited web pages into a topic hierarchy, using this model to re-rank future search results. Also, a particularly straightforward yet

effective search interaction personalization approach is PClick, proposed by Dou et al. [91]. This method involves promoting URLs previously clicked on by the same user for the same query. The user representation model we present in this work could be easily used in any of these personalization techniques.

2.3.2 Gestures for Relevance

Traditional evaluation techniques relied on classical methodologies that use query sets and relevance judgments. More recently, a number of different interaction behaviors have been taken into consideration in the prediction of search user satisfactions including both coarse-grained features (e.g. clickthrough based features in [99]) and fine-grained ones (e.g. cursor position and scrolling speed in [100]). Mouse movement information like scroll and hover have proven to be valuable signals in inferring user behavior and preferences [101, 102, 103], search intent [104], search examination [105] and predicting result relevance [106]. However, none of these studies tried to extract mouse movement patterns and adopt them to predict search satisfaction. Arapakis et al. [107] extracted mouse gestures to measure within-content engagement. Lagun et al. [108] introduced the concept of frequent cursor subsequences (namely motifs) in the estimation of result relevance.

User action sequences have been used to predict user satisfaction [109], graded satisfaction [110] and to study search engine switching behavior [111, 112]. Sequential user actions have also been used to explore developing search trails composed of query sequences for enhancing search support [113, 86]. Liu *et al.* [114] estimate the utilities of search results and the efforts in search sessions with motifs extracted from mouse movement data on search result pages (SERPs).

2.3.3 User Satisfaction

The concept of satisfaction was first introduced in IR researches in 1970s according to Su et al. [115]. A recent definition states that ”satisfaction can be understood as the fulfillment of a specified desire or goal” [116]. However, search satisfaction itself is a subjective construct and is difficult to measure. Some existing studies tried to collect satisfaction feedback from users directly. For example, Guo *et al.*

's work [100] on predicting Web search success and Feild et al.'s work [117] on predicting searcher frustration were both based on searchers' self-reported explicit judgments. Differently, other researchers employed external assessors to restore the users' search experience and make annotations according to their own opinions. For example, Guo et al.'s work [99] on predicting query performance was based on this kind of annotations.

Recently, simplistic user feedback signals have been used to gauge user satisfaction. For instance, it has previously been shown that clicks followed by long dwell times are correlated with satisfaction [118]. Hassan et al. [119] propose to use query reformulation as a negative indicator of search success and thus satisfaction and show how an approach based on query features outperforms an approach based on click features, with the best performance being achieved by a combination of the two. Kim et al. [120] consider three measures of dwell time and evaluate their use in detecting search satisfaction. Lagun *et al.*[121] consider scroll and viewport features for predicting satisfaction in mobile search.

The background on general IR, understanding user behavior and user interactions provided thus far provides readers with the grounds necessary to conceptually understand the work proposed in this thesis. This thesis additionally builds up on some of the recent advancements in the different areas of Machine Learning, some of which we discuss in the next section.

2.4 Algorithmic approaches

Part of the thesis builds upon and extends various machine learning algorithms, from different areas. We next briefly discuss background work for each.

2.4.1 Distributional Semantics for IR

While many word embedding models have been proposed recently, the Continuous Bag-of-Words (CBOW) and the Skip-Gram (SG) architectures proposed by Mikolov et al. [122] are arguably the most popular. Word embeddings have also been studied in IR contexts such as term reweighting [123], cross-lingual retrieval [124, 125] and short text similarity [126]. Beyond word co-occurrence, recent stud-

ies have also explored learning text embeddings from clickthrough data [127], session data [128] and for query prefix-suffix pairs [129]. While most of these works aim at learning richer embeddings, we instead focus on using existing embeddings in a novel way. We explore the use of embeddings by proposing a novel distance metric which takes into account the task information.

2.4.2 Nonparametric Priors

The Dirichlet Process (DP) [130] is a prior over a countably infinite set of atoms, and is popularly used as a prior for mixture models (DP Mixture Model) in applications, where the number of clusters is difficult to provide as a parameter. The Chinese Restaurant Process (CRP) [131] provides a generative description for the Dirichlet Process, and is useful for designing sampling algorithms for DP mixture models. Recently, online variants of CRPs have been proposed [132] which make it possible to model streaming data with CRPs. Socher *et al.* [133] proposed a method to cluster non-exchangeable data that combines the advantages of nonparametric and spectral methods. dd-CRPs have also shown promising results for person discovery in videos [134], POS induction [135] and for modelling influence in social media [136].

2.4.3 Hierarchical Models

Rich hierarchies are common in data across many domains, hence quite a few hierarchical clustering techniques have been proposed. The traditional methods for hierarchically clustering data are bottom-up agglomerative algorithms. Probabilistic methods of learning hierarchies have also been proposed [137, 138] along with hierarchical clustering based methods [139, 140]. Most algorithms for hierarchical clustering construct binary tree representations of data, where leaf nodes correspond to data points and internal nodes correspond to clusters. There are several limitations to existing hierarchy construction algorithms. The algorithms provide no guide to choosing the correct number of clusters or the level at which to prune the tree. It is often difficult to know which distance metric to choose. Additionally and more importantly, restriction of the hypothesis space to binary trees alone is

undesirable in many situations - indeed, a task can have any number of subtasks, not necessarily two. Past work has also considered constructing task-specific taxonomies from document collections [141], browsing hierarchy construction [142], generating hierarchical summaries [143]. While most of these techniques work in supervised settings on document collections, our work instead focused on short text queries and offers an unsupervised method of constructing task hierarchies. Finally, Bayesian Rose Trees and their extensions have been proposed [144, 145, 137] to model arbitrary branching trees. These algorithms naively cast relationships between objects as binary (0-1) associations while the query-query relationships in general are much richer in content and structure.

2.4.4 Task Extraction Algorithms

The analysis on searcher's multi-tasking behavior presented later in the thesis assumes that search log data is pre-tagged with task information. While we propose new subtask and task hierarchy extraction algorithms in Chapter 4 and 5 respectively, we make use of an existing state-of-the-art task extraction algorithm proposed by Wang *et al.* [10] to extract *on-task* queries, i.e., pre-tag search queries with task information. In this section, we briefly describe the algorithm in detail.

Given query sequences within sessions, Wang *et al.* proposed that search tasks are identified by clustering queries into tasks by find the strongest link between a candidate query and queries in the target cluster (*bestlink*). This is achieved by making use of a structural learning method with latent variables, i.e., latent structural SVMs, to utilize the hidden structure of query inter-dependencies to explore the dependency among queries within the same task.

Given a query sequence $Q = q_1, q_2, ..., q_M$, a feature vector for the task partition y is specified by the hidden best-link structure h as $\psi(Q, y, h)$. Based on $\psi(Q, y, h)$, the bestlink SVM is a linear model parameterized by w, and predicts the task partition by,

$$(y*, h*) = argmax_{y,h} w^T \psi(Q, y, h) \tag{2.1}$$

where Y and H represent the sets of possible structures of y and h respectively. $y*$

becomes the output for cross-session tasks and $h*$ is the inferred latent structure. Based on the best-link structure, $h(q_i, q_j) = 1$ if query q_i and q_j are directly connected in h; and otherwise, $h(q_i, q_j) = 0$, with the added clause that a a query can only link to another query in the past, or formally, $\sum_{i=0}^{j-1} h(q_i, q_j) = 1 \ \forall j \geq 1$. The feature vector for any particular task partition y is defined over the links in h as,

$$\psi(Q, y, h) = \sum_{i,j} h(q_i, q_j) \sum_{s=1}^{S} \phi_s(q_i, q_j) \tag{2.2}$$

where a set of symmetric pairwise features $\phi_s(q_i, q_j)$ is given to characterize the similarity between query q_i and q_j. Given a set of query logs with annotated tasks, the feature vector design and the directed linkage structure of h can be inferred in an SVM setting. A detailed overview of the approach can be found in Wang *et al.* [10].

2.4.5 Tensor Factorization

Chapter 6 of this thesis presents a user modelling approach based on task information wherein we leverage recent advancements in tensor factorization. This section briefly introduces tensors and lays background needed to better understand the contributions presented in Chapter 6.

In recent years, researchers have started to realize that many phenomena are inherently multi-way. In such a case, tensors are more natural data representations than matrices - stacking the data in a matrix results in loss of information. Tensor decompositions often have better uniqueness properties than matrix decompositions, which makes that they are often easier to interpret. Multilinear algebra is richer than vector/matrix algebra, which means that more information can be extracted. Roughly speaking, generalizing different properties of the matrix SVD leads to different tensor decompositions.

A tensor is a multidimensional array. More formally, a N-way tensor or N-th order tensor is an element of the tensor product of N vector spaces each of which as its own co-ordinate system. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. The

order of a tensor is the number of dimensions, also known as *modes*. A third order tensor can be represented as $T \in \Re^{I_1 \times I_2 \times I_3}$ with each element of the tensor denoted as $t_{i,j,k}$ with $i \in (1, I_1)$, $j \in (1, I_2)$ and $k \in (1, I_3)$.

Existing tensor factorization methods vary in their sensitivity to noise in the tensor, their tolerance of non-orthogonality and in their convergence properties. A Tucker Decomposition of a tensor $T \in \Re^{I_1 \times I_2 \times I_3}$ is a decomposition of T of the form

$$T = D \circ A \circ B \circ C \tag{2.3}$$

The symbol $\circ$ represents the vector outer product. This decomposition was introduced by [146, 147]. It is not unique. For instance, if A is post-multiplied by a square nonsingular matrix F, then this can be compensated by replacing D by $D \circ F^{-1}$. Part of the degrees of freedom can be used to make A, B and C column-wise orthonormal. One particular constrained version of the Tucker decomposition can be obtained by computing A as the matrix of left singular vectors of an (IJK) matrix in which all the columns of T are stacked one after the other; B and C are obtained by working with the rows and mode-3 vectors, respectively. Tucker approximation is useful for dimensionality reduction of large tensor datasets. The actual data analysis can then be carried out in a space of lower dimensions. Tucker approximation is also important when one wishes to estimate signal subspaces from tensor data.

Another approach for tensor decomposition is the PARAFAC tensor decomposition [148]. By PARAFAC, the input tensors are transformed into Kruskal tensors, a sum of rank-one-tensors. Formally, the tensor $T \in \Re^{I_1 \times I_2 \times I_3}$ is decomposed into component matrices $U \in \Re^{I_1 \times d}$, $T \in \Re^{I_2 \times d}$ and $S \in \Re^{I_3 \times d}$ and d principal factors λ_i in descending order. Via these, tensor T can be written as a Kruskal tensor by:

$$T \approx \Sigma_{k=1}^{d} \lambda_k \cdot U^k \circ T^k \circ S^k \tag{2.4}$$

where λ_k denotes the k-th principal factor. The goal is to compute a decomposition

with d-components that best approximates our tensor T, i.e., to find

$$min_{\widetilde{T}} \|T - \widetilde{T}\| \tag{2.5}$$

such that

$$\widetilde{T} = \Sigma_{k=1}^{d} \lambda_k \cdot U^k \circ T^k \circ S^k \tag{2.6}$$

The above objective could be solved using the Alternating Least Squares (ALS) approach [149] - having fixed all but one matrix, the problem reduces to a linear least-squares problem.

Readers interested in details about multilinear subspace learning for tensor data and a survey of tensor factorization approaches are referred to De *et al.* [150] and Lu *et al.* [151].

Part I

Understanding and Extracting Tasks

Chapter 3

Understanding Search Behavior

3.1 Introduction

Search engine users' information needs span a broad spectrum [70]. While simple needs, such as homepage finding, can mostly be satisfied via a single query, users may also issue a series of queries to collect, filter, and synthesize information from multiple sources to solve a task. Given the inherent diversity in information needs, users engage with search systems in varied ways.

Search sessions have been exploited in previous work on information search, as being the major focus for most analysis of search behavior. The context of search activities within the current session has been used to build richer models of interests and improve how the search system interprets the user's current query. Session context has been used for modeling query and click sequences [57, 58], to disambiguate current search query [59], to build topical profiles for future interest prediction [8], to improve search quality [152, 153] to quantify struggling users [154], for understanding learning and expertise development [51] and for detecting atypicality in user behavior [155].

While search sessions are an important and convenient source for analysis, we contend that this conceptualization of sessions as focal units of analysis makes certain assumptions that are quite untenable in the general case. First, there exists no theoretical basis for bounding search sessions, as it is largely a data-driven subject. Previous research on the topic have adopted a time-out based strategy to

bound search sessions [50, 156, 157]. However, it remains to be understood if such time-out based techniques have strong external validity across search contexts. Second, and most importantly, evidence from analysis of search logs show that users do indeed search for multiple unrelated topics within a single search session.

Further, and as a direct result of the increasingly complex informational environment around us, users are increasingly engaged in multitasking and informational task switching behaviors. Multitasking is the ability of humans to simultaneously handle the demands of multiple tasks through task switching [158][159]. While such multi-tasking behavior is becoming increasingly popular especially in the context of online search, many interactive technologies do not provide effective support for managing such multitasking behaviors.

Web search engines offer a typical environment where users perform multiple search tasks across diverse contexts. For example, a programmer searching for solutions to a bug in his code, might take a brief hiatus to listen to some music. The two tasks described here need not be at the same level of importance for the user, nor must they be performed in parallel. While such situations are commonly observed in our daily search behavior, not much is understood about the kind of users who indulge in such multi-tasking behavior or even the extent or nature of such multi-tasking behavior in major search engines. This research gap stems mainly from the difficulty in identifying, quantifying and fully describing multiple task completions from observational data.

In this chapter, we leverage back-end search logs from a large-scale search engine to provide a detailed analysis of multi-tasking behavior for different user groups, and across multiple session sessions over a 30-day period. Our analysis demonstrates the presence of multiple search tasks within single session. We seek to provide evidence that multi-tasking has emerged as a dominant characteristic of online search behavior and that users have varying propensities to indulge in such multi-tasking. Further, we also uncover significant heterogeneities in search topics across single- and multi-task sessions, and across different user groups.

Departing from existing studies on multi-tasking which have used topics of

queries as proxies for tasks, we make use of an explicit search task extraction framework to extract the task information from web search sessions. This allows us to provide richer insights on the prevalence of task multiplicity in search sessions. Making use of real world search logs, we first quantify the extent of multi-tasking behavior in search sessions and show the existence of user groups based on the extent of multi-tasking behavior. Further, we analyse the user groups on a number of search interaction metrics and quantify the differences in these user groups based on how they interact with search systems.

Finally, we investigate user search task behavior for both single- as well as multi-task search sessions and relate it to tasks and topics. Specifically, we analyze user-disposition, topic and user-interest level heterogeneities that are prevalent in search task behavior. Our results show that while search multi-tasking is a common phenomenon among the search engine users, the extent and choice of multi-tasking topics vary significantly across users. We find that not only do users have varying propensities to multi-task, they also search for distinct topics across single-task and multi-task sessions.

3.2 Characterizing Multi-Tasking Behavior

In this chapter, we seek to characterise user behavior in online search sessions based on task specificity and multiplicity. While users generally perform a single task in a single search session, the task process might get interrupted by other competing tasks that become salient in the particular context. Given this backdrop, we contend that it is imperative for the search engine to understand the type of users who might be more prone to multi-tasking within a single session, and also the type of tasks that might be more susceptible to interleaving or interference by competing tasks.

3.2.1 Research Questions

In this chapter, we formulate and propose the following 5 research questions, and offer evidence from a large scale observational dataset on search behavior to answer them.

1. **RQ1**: *To what extent do users multi-task in web search sessions?*

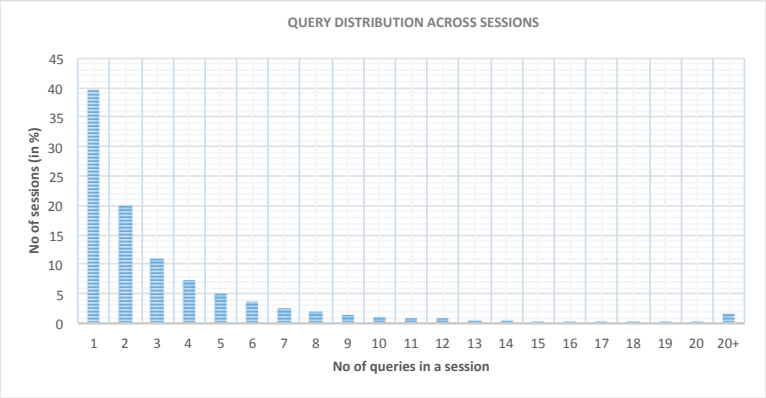


Figure 3.1: The variations of number of the number of queries in a session.

2. **RQ2**: ***Quantifying user heterogeneity:*** *Is there evidence of user-level heterogeneity based on task behavior? i.e. do different users behave differently in terms of their multi-tasking behaviors?*

   (a) **RQ2.1**: *How does the search task effort vary across different user groups?*

   (b) **RQ2.2**: *Is there an association between users' choice of search topics and task multiplicity?*

3. **RQ3**: ***Quantifying topic-category heterogeneity:*** *Is there a topic-category level heterogeneity across single- and multi-task sessions, for different user groups?*

3.2.2 Data Context

We use back-end search logs for a subset of users from a major US-based search engine for a period of 30 days from May 1, 2015 to May 31, 2015, and choose a random sample of over 2 million users where each user is identified by a unique user ID derived from their IP address. Over the 30-day period, users participated in a total of over 200 million search sessions comprising one or more search queries, as illustrated in Fig. 3.1, wherein sessions are identified based on an inactivity period of 30 minutes. In order to avoid biasing the results by inactive users, we filter out users from our dataset who participate in $\leq$50 sessions, and focus instead on the more active user population.

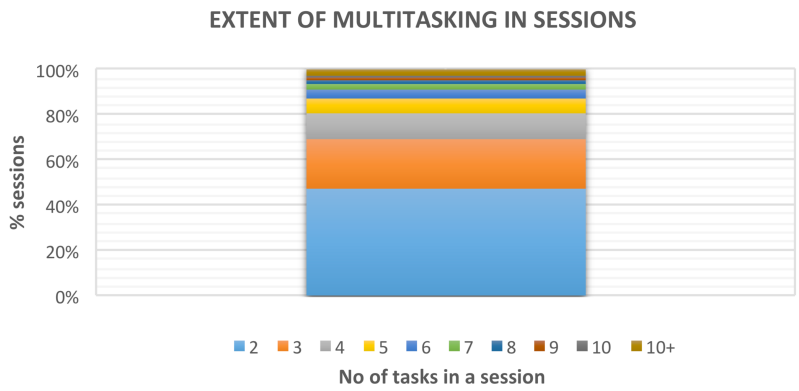


Figure 3.2: Quantifying the extent of multi-tasking in search sessions: .

3.2.3 Task Extraction

For our analysis, we make use of the Latent Structural SVM framework [10] for task identification. Given query sequences within sessions, search tasks are identified by clustering queries into tasks by find the strongest link between a candidate query and queries in the target cluster (*bestlink*). This is achieved by making use of a structural learning method with latent variables, i.e., latent structural SVMs, to utilize the hidden structure of query inter-dependencies to explore the dependency among queries within the same task. The algorithm is described in more detail in section 2.4.4 of the previous chapter.

3.3 Quantifying the Extent of Multitasking

While it is well known that online search sessions often tend to have interleaving of multiple tasks, an understanding of the multi-tasking heterogeneities at a user level and/or a search session level has been largely ignored. Specifically, we aim at quantifying, first, the prevalence of multi-tasking behavior in online search sessions (i.e. *how common is multi-tasking?*), and second, the extent of multi-tasking behavior in multi-task sessions (i.e. *how many tasks on average are there in multi-task search sessions?*).

Fig.3.2 illustrates the prevalence and extent of multi-tasking behavior in the sessions which have more than one tasks. In our dataset, we find close to 90 million search sessions which have 2 or more completed tasks. Among these 90 million search sessions, there is a varied distribution of task multiplicity as described by

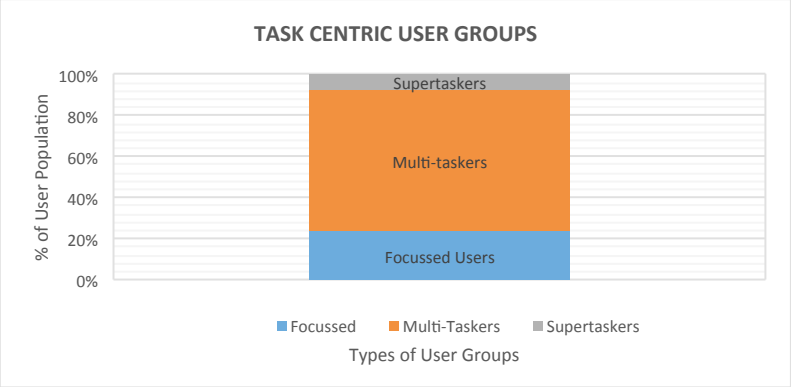


Figure 3.3: User groups based on multi-tasking behaviors.

Fig. 3.2. Specifically, we observe that while over 60% of the sessions have 2 or 3 tasks, about 20% of the sessions have 5 or more tasks. On average, each user participates in 76 sessions in which they performs an average of 2 tasks.

We next extend our multitasking analysis to investigate user level differences in multi-tasking behavior.

3.4 Uncovering User Level Heterogeneity

In this section, we seek to uncover the presence of user idiosyncrasies in multi-tasking behavior in search sessions. Specifically, we attempt to understand the proportion of sessions per user that are single tasked vs. multi-tasked. Consequently, we seek to uncover any underlying categorizations among the users based on the extent of their multi-tasking behavior. Specifically, we wish to answer questions like: *Can we identify and classify groups of users who demonstrate similar proportions of multi-tasking behavior?*. Uncovering such user groups would pave the way for the search engine to provide better personalized search assistance based on the group-level features and characteristics. We next describe the user groups obtained.

3.4.1 Uncovering User Groups

In an attempt to uncover different groups of users based on their multi-tasking habits, we look at the user-level heterogeneities in search sessions. Specifically, we categorize users based on the average number of tasks completed by the user across all sessions in the 30-day period. We observe that a sizable number of users (i.e. more than 20%) perform just a single task on average across their session his-

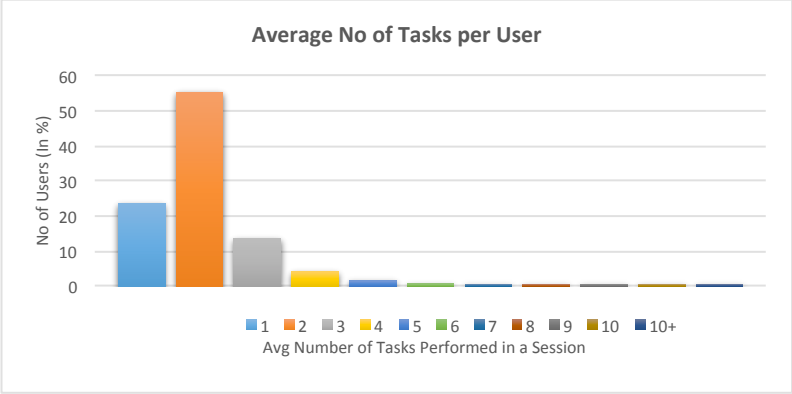


Figure 3.4: Investigating User Level variations in Multi-tasking.

tory. We call such users ***focussed users***, as their search behavior is focussed on a specific task, free from interference of competing tasks. On the other extreme, we also find a small group of users who perform 4 or more tasks on average across their session history. We call such users ***supertaskers***, who perform several tasks within a single session. We categorize all the other users as ***multi-taskers*** who completed more than 1 but less than 4 tasks on average in their session history.

The density of users across each of the three groups have been better depicted in 3.3. Moreover, we show the task multiplicity distribution for our user sample in 3.4. Interestingly, from Fig.3.3 and Fig.3.4, we observe that most users are not focussed in the search behavior, and tend to complete at least 2 tasks within a session. This is not unsurprising, given that one of the tasks could be the primary (e.g. search for solution to a programming bug on the Internet) or important task, while the others might be ancillary tasks (e.g. listen to music, check weather updates). Table 3.1 provides a list of sample queries executed by users across the three user groups.

3.4.2 Characterizing Effort Across User Groups

The presence of competing or interfering tasking within a single session could accentuate or attenuate the search effort expended by the users. Specifically, we wish to understand the relationship between task multiplicity and total effort expended by the users (i.e. *do users who multitask more(less) expend more effort than users who multitask less(more)?*. In the context of the current study, following part work around search effort, we operationalize search effort using the query time, the aver-

User Type	Example Queries from an exemplary session
Focussed User	”test guide.com”, ”CNA Practice Test”, ”CNA State Board Exam”, ”CNA Testing Schedule and Locations”, ”CNA State Board Practice Test”, ”CNA Practice Test 2014”, CNA 50 Questions Test”, ”Free GED Practice Test 2014”
Multi-Tasking User	”Gravity FSX 2.0”, ”Full Suspension Mountain Bikes”, ”Walmart Cards”, ”Walmart Instant Card Application”, ”Gravity FSX 2.0 price”, ”full suspension bike sale”
Supertasking User	”hairstyles for women over 50”, ”thin wavy hairstyles for women”, ”facebook”, ”fb sign in”, pulled pork crock pot recipe easy”, ”Slow-Roasted Pulled Pork”, ”barefoot contessa”, ”miley cyrus hair styles”, ”hairstyler.com”

Table 3.1: Example query sessions from the different user groups.

age length of queries and other metrics as described below.

We analyse search effort across the three user groups identified in the study. We characterise search effort using 4 different metrics:

1. **Time to First Click (TTFC)**: measures the time elapsed before the user clicks the first link on the query result page. A longer TTFC is an indication of user surprise or confusion with the search results, and hints at a more extensive cognitive elaboration process as the user decides which link to click on.

2. **Time to Last Click (TTLC)**: measures the time elapsed before the user clicks the final link in her search session. The TTLC is a more direct measure of search effort expended by the user within a particular session. A higher TTLC could indicate that the user disatisfaction with the early results provided by the query results, or a heightened motivation on part of the user to search more about the particular topic of interest.

3. **Page Click Count (PCC)**: quantifies the total number of clicks observed on the SERP.

4. **Pagination Click Count (PgCC)**: quantifies the amount of pagination activity observed. Higher PgCC values indicate users examining results from beyond the first page. Both PCC & PgCC metrics are direct measures of the search effort put in by the user, and are hence good proxies to capture the

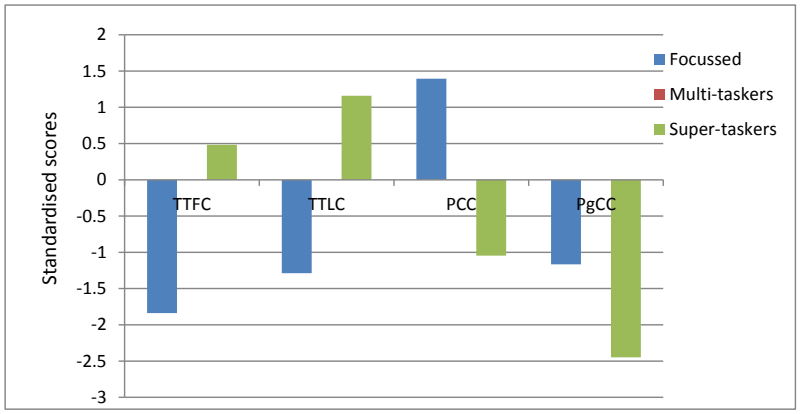


Figure 3.5: Differences in user groups quantified via effort based metrics. The scores reported are deviations from the Multi-taskers group which is held as baseline. All numbers are standard scores (Z-scores).

different facets of search effort.

We compute each of these metrics across each of the user groups as defined earlier, and highlight our findings in Figure 3.5. Our analysis indicates that supertasking users have a much higher TTFC and TTLC scores, but a lower PCC score than the focussed and multi-tasking groups. This supports our conjecture that most supertaskers perform multiple tasks in a master-slave fashion, where they focus bulk of their attention on a focal task, while being periodically distracted by ancillary tasks (e.g. music, weather updates). This periodic distraction causes a decrease in attention span on the focal task, which is manifested by a decrease in click count, and an increase in click delays on the focal task. We do not, however, find any noticeable difference in the PgCC scores across the groups.

Given the user heterogeneity in session level multi-tasking, an important question arises as to how these multi-tasking behavior links to user's interest profiles. In the next section, we investigate such behavior in terms of topical profile of users and quantify the topic-wise extent of search multi-tasking.

Time	Query	SessionID	TaskID	Topic
05/29/2012 14:06:04	adele songs	1	1	Arts
05/29/2012 14:11:49	wedding venue	1	2	Society
05/29/2012 14:12:01	video download	1	3	Arts
05/29/2012 14:06:04	Obama care	2	4	News
05/29/2012 14:11:49	running shoes	2	5	Shopping
05/29/2012 14:12:01	sports shoes	2	5	Shopping
05/29/2012 14:22:12	wedding cards	2	2	Society

Table 3.2: Sample search sessions

3.5 Uncovering Behavioral Heterogeneities in Search Behavior

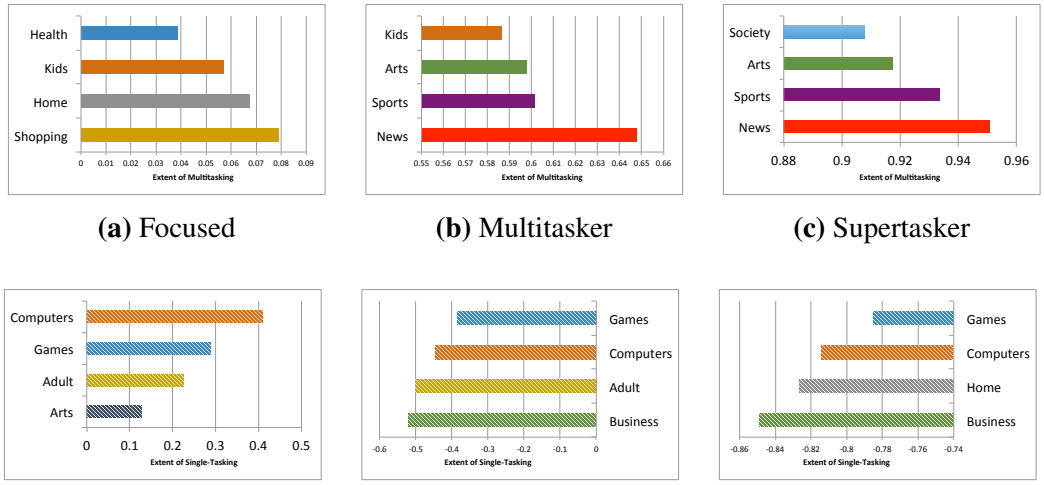
In characterizing these search tasks across sessions, we consider the possibility of three distinct forms of heterogeneity inherent in the search-task behavior. First, there could be *user-disposition* level heterogeneity wherein some users have a higher propensity to multi-task when searching for information, than other users. Second, there could be *topic* level heterogeneity wherein searchers have a higher (or lower) propensity to multi-task when searching information for specific kind of topics. Third, and finally, there could be *user-interest* level heterogeneity wherein users might have a higher or lower propensity to multi-task when searching for topics they are most or least interested in.

While recent work has highlighted the prevalence of multi-tasking behavior in online search [70, 16, 80], not much effort has been expended at fully characterizing online search tasks with an emphasis on such user- and topic-level differences. This section focuses on such differences. Specifically, we find that while most users (>50%) choose to multi-task in their search sessions, there exists significant differences in their choice of topics between single-task and multi-task sessions.

Through our analyses, we offer the following three insights:

1. **Users' preference towards multitasking(3.3):** We find evidence that most users multi-task when searching for information with over 50% users completing more than 2 tasks within a single search session, and a minority of users even completing more than 5 tasks within a single session.

2. **Topic level heterogeneity(3.5.1):** For certain type of topics, users prefer to



(a) Focused **(b)** Multitasker **(c)** Supertasker

(d) Focused **(e)** Multitasker **(f)** Supertasker

Figure 3.6: Top topics prone to multi-tasking (Top) and single-tasking (Bottom) across different user groups.

multi-task (e.g. kids, news, shopping etc.), while for certain others, users prefer to single-task (e.g. computers, games, adult etc.).

3. **User-interest level heterogeneity(3.5.2):** Users have different preferences towards multitasking depending on their level of interest in the specific search topic (e.g. some groups of users prefer to search about most-interested topics in single-tasking sessions and least-interested topics in mult-tasking sessions).

3.5.1 User-disposition and Topic Level Heterogeneity

Recent work on the topic of search multi-tasking has shown that a majority of users perform two or more tasks within a single search session [16]. Consistent with these studies, our analysis also uncovers that close to 80% of users perform two or more tasks within a single session, with a minority of users even performing 5 or more tasks within the single session, as illustrated in Figure 3.3. We term these three discernible classes of users based on their frequency of multitasking behavior viz. focused (i.e. 1 task per session), multitaskers (i.e. 2-5 tasks per session) and supertaskers (i.e. $>$5 tasks per session). Having established that users vary on their disposition to single-task and multi-task, we now delve deeper into understanding whether users multi-task to varying extents depending on their search topics.

To obtain such a topic representation for this study, we labeled each document with a vector of probabilities of categories from the top two levels of the Open Directory Project (ODP) hierarchy using a text-based classifier. Each documents vector was restricted to the three most probable classes. The classifier has a micro-averaged F1 value of 0.60 and is described more fully in [8]. The most prominent topic among the top 3 returned results per query was used as the final tagged topic for that query.

We analyse topic level heterogeneity by investigating the level of multi-tasking in sessions filtered by topics. Our results are illustrated in Figure 3.6 wherein we highlight the top 4 most prevalent topics across multi-tasking and single-tasking sessions (top to bottom panels), for all three category of users (left to right panels). The length of the bars in each of the charts in the Figure 3.6 highlights the extent of multitasking (top panels) and the extent of single-tasking (bottom panels). The extent of multi-tasking is defined as $\frac{N_M - N_S}{N_{total}}$, which measures the difference between the proportion of times the topic featured in a multi-tasking session (N_M) and the proportion of times the topic featured in a single-tasking session (N_S). Conversely, the extent of single-tasking was calculated as the difference between the proportion of times the topic featured in a single-tasking session and the proportion of times the topic featured in a multi-tasking session.

We find that focused users primarily multi-task for topics related to shopping, home, kids, health and recreation. However, both multi- and super-taskers have a shared preference for multi-tasking on topics related to news, sports and arts. We also observe that focused users prefer to single task when searching for topics related to computers, games, adult and arts categories, while multi-taskers and super-taskers do not prefer to single-task when searching for their preferred topics. This is reflected by the negative scores on the extent of single-tasking in the bottom panel of Figure 3.6. These findings confirm our intuition that indeed certain topics are more prone to multi-tasking (e.g. news, sports) while others (e.g. computers, adult) usually witness single tasking sessions.

	Focussed		Multi-taskers		Super-taskers	
	Single-Tasking	Multi-Tasking	Single-Tasking	Multi-Tasking	Single-Tasking	Multi-Tasking
Most Interested Topics	0.593	0.407	0.310	0.690	0.105	0.895
Least Interested Topics	0.458	0.542	0.249	0.751	0.081	0.919

Table 3.3: Relating User's Mono/Multitasking Nature with their interest profiles.

3.5.2 User-interest Level Heterogeneity

We next investigate whether users exercise any specific search preference when searching for topics that are of high vs. low interest to them. To analyze this, we compute the frequency of most and least searched topic categories from the search history of users in each of the three user groups viz. focused, multi-taskers and super-taskers [1]. Following this, we analyze their search behavior during single-tasking and multi-tasking sessions to investigate the distribution of high and low interest topic categories across these sessions. The results from this analysis are described in Table 3.3, and highlight that users exercise distinct preferences towards the extent of their multi-tasking nature in search sessions for high vs. low interest topics.

Our results show that multi-taskers and super-taskers prefer to multi-task for a large majority of their search sessions (i.e. almost always $>$70%), irrespective of whether they are searching for high or low interest topics. In contrast, however, focused users prefer to search for high interest topics in single-tasking sessions (i.e. 59% of the time), and low interest topics in multi-tasking sessions (i.e. 54% of the time) in the small portion of their multi-task sessions.

Analyzing such heterogeneities in online search behavior lends us a better understanding of how users interact with search systems when performing different tasks. The findings from this study offer valuable insights into the search strategies employed by online users on search engines.

3.6 Implications & Discussion

The research presented in this chapter is among the first to analyze and quantify user centric multi-tasking behavior in web search sessions using large-scale and

[1] Note that this is different from the identification of top topics in the previous section which were identified at a session-level and not at a user-level.

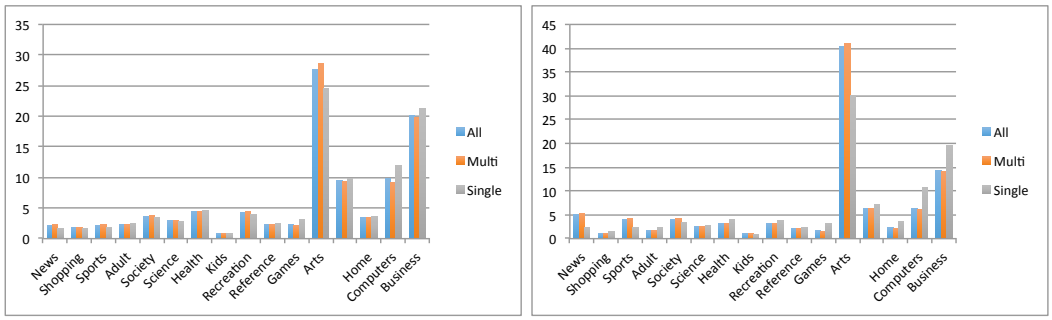


Figure 3.7: Topical distribution of queries for Single tasking & multi-tasking sessions for Multi-taskers (left) & super-taskers (right).

objective search logs. Specifically, we emphasize (Sec 3.4.1) that while most users on search engines are multi-task users performing 2 or more tasks within a single search session, there exist a sizable proportion of users who are more focussed and mostly mono-task. We also provide evidence of "Supertaskers" who perform onwards of 4 tasks within a single session. This widespread prevalence of task multiplicity makes it imperative for search engines to refocus their personalization and recommendation strategy towards a task-oriented view. For example, if search engines can fully identify and characterize the number and types of tasks performed by a given population of users on the engine, they could potentially optimize the session to better fit specific user- and task-based needs, while also making potential task-recommendations to reduce the search effort, as quantified in this chapter.

As highlighted by the varying search effort metrics in Sec 3.4.2, the different user groups indeed interact with the search results differently and hence motivate the need for incorporating such differences in multi-tasking behavior of users while personalizing search experiences.

Yet another finding we wish to highlight through our study is the characterizing of multiple tasks into a combination of a single primary and multiple ancillary tasks. Our task effort scores provide preliminary evidence to suggest that such categorization of multiple tasks into a task-hierarchy might indeed be plausible. Such insights are useful for search engines in that they could reduce task-transition delays and make design improvements to reduce cognitive loads in such multi-task sessions.

Finally, we performed a deeper task-level analysis to uncover possible topical

differences among tasks that occur frequently in multi-task sessions, versus those that occur frequently in single-task sessions (i.e. *Are tasks & topics in single-task sessions qualitatively different than those prevalent in multi-task sessions?*. Understanding such task-level differences provide a stepping stone towards understanding specific task-characteristics that might make it more or less susceptible to interference and/or distraction.

These insights could help in making more accurate task-predictions within a single search session, as well as across multiple sessions for a given user. With a steadily improving understanding of task and search behavior online, we envision a day when the search engine would be able to infer user-,task- as well as session-level characteristics based on just the first query issued by the user and user's multi-tasking habits, and personalize their search experience accordingly.

Chapter 4

Exploiting Distributional Semantics with Nonparametric Priors for Extracting Sub-Tasks

4.1 Introduction

User initiated search is often motivated by their informational need required to achieve a goal, or a task such as booking travels, buying a house, etc. Search engines play an important role in not just facilitating such information discovery but also in guiding users towards completing specific tasks, by offering search results in a fashion which assists users in completing their tasks.

While existing search engines are adept at handling simple information seeking needs composing simple tasks, users get little or no help when their information need transcends this boundary. This major limitation majorly stems from search engine's treatment of search tasks as structure-less clusters which inherently lack insights about the presence or demarcation of subtasks associated with individual search tasks [70, 10, 6] . A more naturalistic viewpoint would involve considering complex search tasks as being decomposed into more focused subtasks. For instance, a complex task like planning a wedding involves many different sub-tasks like buying bridal dresses, deciding on wedding themes, searching for invitation card designs etc., each of which is a subtask which a user intends to accomplish by

issuing a set of related queries.

Clearly, identifying and analysing subtasks becomes an extremely important activity for search engine providers in their effort to improve user experience on their platforms. Subtask identification turns out to be a complex problem for three reasons. First, the number of sub-tasks in a given task is not a parameter than can be explicitly defined and is generally and strongly task-dependent. Second, while similar sounding queries like ”wedding planning checklist” and ”wedding dress” belong to the same task, they inherently represent different sub-tasks. This necessitates the use of advanced distancing techniques, beyond the usual bag-of-words or TF-IDF approaches to identify subtask clusters, which are coherent and homogenous. Finally, it is often non-trivial to homogeneously demarcate the sub-tasks due to the strong overlap in the informational needs embodied by the different subtasks which makes the process of identifying coherent subtasks all the more important.

In this chapter, we focus on extracting coherent subtasks from a given query collection of *on-task* queries. We exploit the benefits offered by modeling bayesian nonparametrics jointly with distributional representations and propose a nonparametric models to extract subtasks. Specifically, we propose a novel generative model based on subtask coherence estimates, which is not restricted by a fixed number of sub-task clusters, and assumes an infinite number of latent groups, with each group being described by a certain set of parameters. We specify our non-parametric model by defining a Distance-dependent Chinese Restaurant Process (dd-CRP) prior and a Dirichlet multinomial likelihood [160]. The non-parametric model is enriched by working in the vector embedding space which uses a word-embedding based distance measure to encode query distances for efficient sub-task extraction. Further, we formally define the notion of subtask affinity, which helps us quantify the semantic cohesiveness and coherence of a given subtask, based on which we propose a novel likelihood function which encodes the coherence estimates. We hypothesize that a coherence aware subtask extraction technique enables us to extract more cohesive and homogeneous subtasks. We validate our proposed method using a number of experiments, including both qualitative and quantitative

evaluations to demonstrate that our proposed *TE-coh-ddCRP* model is able to extract coherent tasks. We additionally show how a search engine might benefit from the proposed subtask discovery, by investigating user effort expended in the different subtasks.

4.2 Problem Formulation

Our definition of search tasks follows from previous work [13] which identified tasks as search missions and goals. Often search tasks involve many distinct, but related aspects which warrant the need for issuing different sets of queries over a number of sessions in order to fulfil the multi-aspect information needs. It is mostly the case that these independent information needs arise from an overall complex search goal or task. Following past work, Complex Search Tasks which can be defined as a multi-aspect or a multi-step information need consisting of a set of related tasks [13, 9]. A complex search task could be broken down into smaller multi-step or multi-aspect sub-tasks that represent atomic informational needs, for which it is trivial for users to issue satisfying queries.

In this chapter, we explicitly focus on complex search tasks and intend to extract the different subtasks associated with them. Given a collection of *on-task* queries, i.e., queries belonging to the same overall task, our goal in this chapter is to extract the subtasks. It is important to note that while the queries are observed, the inherent sub-tasks and their numbers are latent. Indeed, tasks differ in complexities and different tasks would have different number of subtasks. We present an approach to subtask extraction which makes use of a predefined collection of *on-task* queries. We next describe the process by which we collect such *on-task* queries, and use this process throughout the rest of the paper.

We draw from a number of existing work on bayesian nonparametric models, task extraction techniques and nonparametric subtask extraction methodologies. In this section we go through the background material in detail and define constructs which will be used throughout the paper. In Section 7.2.1, we describe an existing task extraction technique used to extract "*on-task*" queries.

plan wedding	buy shoes	get insurance	listen to music
wedding theme text	running shoes	insurance rates	song download site
wedding printables	sports wear shoes	buy insurance	free music download
wedding dresses gowns	men's shoes	cheap insurance	latest soft rock
wedding planners	women's shoes	car insurance rates	streaming music free
inexpensive wedding ideas	shoe stores online	insurance agents	blues reggae songs
wedding insurance	shoe discounts	pet insurance	online radio music

Table 4.1: Sample search tasks and associated queries

4.2.1 Extracting "On-Task" Queries

Prior to extracting subtasks, we first need to extract the set queries which belong to the same overall complex task. In order to extract *on-task queries*, we make use of the Latent Structural SVM framework [10] for task identification. We run the task extraction algorithm as described in section 2.4.4 of Chapter 2 on search logs to extract all queries belonging to the same task. Such a query collection is henceforth referred to as "*on-task queries*". Table 4.1 provides example *on-task* queries extracted from four different tasks. As can be seen from the queries constituting different tasks, there are few clear subtasks embedded in those tasks. The given collection of *on-task* queries would be used on a per-task basis to extract subtasks for any given task.

4.2.2 Non-parametric Subtask Extraction

Currently there are several task extraction techniques [9, 10, 70, 6] but the idea of subtask extraction has not yet received considerable attention in the research community. In this section, we provide an overview of a simple bayesian non-parametric approach for sub-task exatraction , which serves as a starting point for our proposed model which is described in detail in the next section (4.3).

Consider a collection of queries (Q) issued by searchers trying to accomplish certain complex search task. Our goal is to extract the different sub-tasks from such a collection of on-task search queries. Since the complexity of different search tasks vary based on the task, the number of subtasks associated with them cannot be explicitly specified beforehand.

The recently proposed distance dependent Chinese restaurant process (ddCRP) [160], a generalization of the CRP underlying Dirichlet process mixture models

[131], has a number of attractive properties which make it particularly well suited for modeling search tasks and subtasks. By placing prior probability mass on partitions with arbitrary numbers of parts, it allows data-driven inference of the true number of sub-tasks underlying the observed search tasks. In addition, by choosing an appropriate distance function we can enforce those set of queries which help users solve a common information need, to group as a common subtask, and hence, guarantee that all inferred subtasks are coherent.

4.2.3 Chinese Restaurant Processes

The Chinese Restaurant Process (CRP) [131] is a distribution on all possible partitions of a set of objects. The generative process can be described by considering a Chinese restaurant with an infinite number of tables and a sequential process by which customers enter the restaurant and each sit down at a randomly chosen table. After N customers have sat down, their configuration at the tables represents a random partition. Customers sitting at the same table are in the same cycle. In the traditional CRP, the customers sit at an occupied table with a probability proportional to the number of customers already sitting there, or a new table with probability proportional to a scaling parameter α.

The distance dependent-CRP [160] alters the CRP by modeling customer links not to tables, but to other customers. In this distribution, the seating plan probability is described in terms of the probability of a customer sitting with each of the other customers. The allocation of customers to tables is a by-product of this representation. When used in a Bayesian model, the posterior provides a new tool for flexible clustering of non-exchangeable data.

4.2.4 Nonparametric Priors for Modeling Sub-Tasks

We formulate the subtask extraction problem in terms of dd-CRPs and propose a novel generative model for the same. In our sub-task extraction problem, each search task is associated with a dd-CRP and its tables are embellished with IID draws from a base distribution over mixture component parameters. Let z_i denote the query assignment for the ith query, i.e., the index of the query with whom the ith

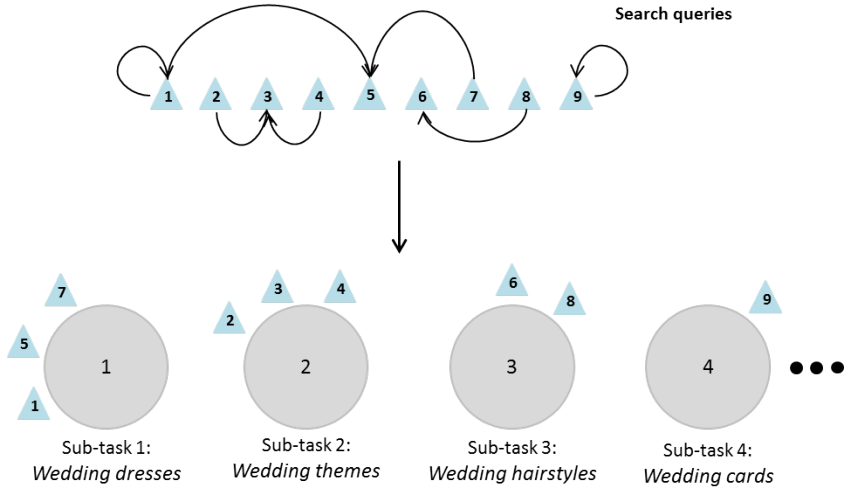


Figure 4.1: Visual formulation of the proposed approach. The tables represent the different subtasks while each traingle represents the search queries. Query assignment leads to subtask assignments.

query is linked. Let d_{ij} denote the distance measurement between queries i and j, let D denote the set of all distance measurements between queries, and let f be a decay function. The distance dependent CRP independently draws the query assignments conditioned on the distance measurements between the queries,

$$p(z_i = j \| D, \alpha) \propto \begin{cases} f(d_{ij}) & \text{if } j \neq i \\ \alpha & \text{if } j = i \end{cases}$$

Here, d_{ij} is an externally specified distance between queries i and j, and α determines the probability that a query links to themselves rather than another query. The monotonically decreasing decay function $f(d)$ mediates how the distance between two queries affects their probability of connecting to each other, i.e., their probability of belonging to the same sub-task. We consider an exponential decay function for all the experiments. Queries are assigned to subtasks by considering sets of queries that are reachable from each other through the query assignments. We denote the induced subtask assignments $t_{(z)}$, and notice that many configurations of query assignments z might lead to the same subtask assignment. Finally, query assignments can produce a cycle, e.g., query 1 linking with 2 and query 2 linking with 1. This still determines a valid subtask assignment: all queries linked in a cy-

cle are assigned to the same subtask. Figure 4.1 provides a pictorial representation of the subtask assignment process. We have described a way of extracting *on-task* queries and briefed a general formulation of the distance dependent CRP. We now describe our subtask extraction framework for modelling search queries into different subtasks. When extracting subtasks, it is important to segregate subtasks which don't solve the same information need into separate subtasks. We present a novel formulation of query clusters which promote task coherence and explicitly encodes subtask coherence as an integral part of the likelihood function. We demonstrate how one might use the posterior distribution of the partitions, given search log data and an assumed generating process based on the distance dependent CRP.

4.3 Extracting Coherent Subtasks

An important characteristic of task and subtask is their coherence, a measure which estimates the cohesiveness of the queries belonging to the task. Often, tasks learned on sparse or noisy query data tend to be less coherent, difficult to interpret, and not particularly useful. Some of these noisy tasks can be vaguely interpretable, but contain one or two unrelated queries, while other subtasks can be practically incoherent with the constituent queries solving completely different information needs altogether. This motivates the need to extract subtasks which are coherent in terms of the task they solve.

We build upon past work on subtask extraction and introduce a novel likelihood function to extract more coherent subtasks. In this section, we first present a way to define and operationalize our notion of task coherence and then present a generative model which incorporates the task coherence aspect when assigning queries to different sub-tasks.

4.3.1 Quantifying Subtask Coherence

When extracting sub-tasks, sub-tasks which help the searcher address a common information need are more desirable. With the help of a task affinity function, we introduce a mechanism which enables the proposed model to extract coherent subtasks. While traditional dd-CRPs enforce similar queries to belong to the same task,

often it might be the case that two queries solve different tasks despite having high similarity scores. In order to bias the subtask extraction algorithm to favor coherent subtask, we introduce the notion of subtask coherence and weight the likelihood of each subtask with its coherence score. More specifically, we define Subtask Affinity as follows:

Definition: Subtask Affinity is a measure indicating the cohesiveness of the query collection in terms of the information need the queries help to address. It is captured by the semantic closeness of the queries associated with the subtask.

By measuring Subtask Affinity, we aim at capturing the semantic variability of queries within this subtask in an attempt to identify how cohesive the query collection represented by the subtask is. To measure task affinity for a given query collection, we propose the use of a novel affinity score based on Pointwise Mutual Information (PMI) [161]. Pointwise Mutual Information has been studied variously in the context of collocation extraction [162] and is one measure of the statistical independence of observing two words in close proximity. We compute affinity score by using PMI scores for each subtask. We split queries into terms and obtain a set of terms corresponding to each subtask, and calculate a subtask's PMI scores using its set of query terms. More specifically, the PMI of a given pair of query terms (w_1 & w_2) is given by:

$$PMI(w_1, w_2) = log \frac{p(w_1, w_2)}{p(w_1)p(w_2)} \tag{4.1}$$

All the probabilities are determined from the empirical statistics of some full standard collection. We employ the AOL log query set for this and treat two query terms as co-occurring if both terms occur in the same search session. For a given subtask (t), we measure subtask affinity as the average of PMI scores for all pairs of the search terms associated with the task node:

$$PMI_t = \frac{1}{|w|} \sum_{i=1}^{|w|} \sum_{j=1}^{|w|} PMI(w_i, w_j) \tag{4.2}$$

$$\psi(q_{t^k(z)}) = PMI_t exp^{\frac{-(PMI_t-b)^2}{c^2}} \tag{4.3}$$

where $|w|$ represents the total number of unique search terms associated with subtask t and PMI_t is the PMI value obtained for subtask t. Since smaller subtasks (2/3 queries) might result higher PMI scores, we introduced a Gaussian weighting term which helps in giving weights to the subtasks based on their PMI scores. The gaussian parameters b indicates the mean subtask coherence at any stage of sampling and c its standard deviation.

4.3.2 Generative Process

The generative process of queries in a subtask is as follows. For each query, we draw a query assignment based on the dd-CRP prior described in Section 4.2.4. Subsequently, the queries are then placed at different subtasks via these assignments, and the query terms then assigned to the queries associated with their tables. Subsets of the queries exhibit a partition structure by sharing the same subtask. The overall link structure specifies a partition: two queries are clustered together in the same sub-task if and only if one can reach the other by traversing the link edges.

Formally, each search task is associated with a distance dependent CRP, and its subtasks are embellished with IID draws from a base distribution over terms or words (G_0). The mixture component for a query depends on the mixture component for nearby queries. G_0 is typically a Dirichlet distribution over distributions of query terms. We define $z^*_{q_{\{1:N\}}}$ to be the first customer (query) to sit at each table (subtask), i.e., those queries who link to themselves. Given a decay function f, distances between queries D, scaling parameter α, and an exchangeable Dirichlet distribution with parameter λ, N M-word queries are drawn as follows,

1. For each query $i \in [1,N]$, draw assignment $z_i \sim dist-CRP(\alpha, f, D)$.
2. For each sub-task, $k \in \{1,....\}$, draw a parameter $\theta^*_k \sim G_0$.
3. For each query $i \in [1,N]$,
   (a) If $c_i \notin z^*_{q_{1:N}}$, set the parameter for the i^{th} query to $\theta_i = \theta_{q_i}$. Otherwise draw the parameter from the base distribution, $\theta_i \sim Dirichlet(\lambda)$.

(b) Draw the ith query, $w_i \sim Mult(M, \theta_i)$.

Since subtask assignment heavily depends on how we specify the distance metric between two queries, we need a way of capturing distances between queries keeping in mind their task relatedness, i.e., two queries belonging to the same subtask should have smaller distance than pair of queries belonging to different subtasks. We next describe our approach in modelling such subtask specific distances between queries.

4.3.3 Quantifying Task Based Query Distances

To capture task specific distances between queries, we propose a novel embedding based distance metric between queries. Word embeddings capture lexico-semantic regularities in language, such that words with similar syntactic and semantic properties are found to be close to each other in the embedding space. We leverage this insight and propose a novel query-query distance metric based on such embeddings. We train a skip-gram word embeddings model where a query term is used as an input to a log-linear classifier with continuous projection layer and words within a certain window before and after the words are predicted. We next describe how we use these query term embedding vectors to define query distances.

For a search task like "planning a wedding", frequent queries include *wedding checklist*, *wedding planning* and *bridal dresses*. Ideally, checklist and planning related queries constitute a different sub-task than bridal dresses. Moreover, given the overall context of weddings, words like *checklist* and *dresses* are more informative than generic words like *weddings*. To this end, we classify each word as **background word** or **subtask-specific word** and use a weighted combination of their embedding vectors to encode a query's vector:

$$V_q = \frac{1}{n_{terms}} \sum_i \frac{n_{q_{t_i}}}{\Sigma_q n_q} V_{t_i} \tag{4.4}$$

where t_i are the terms in the query q, $n_{q_{t_i}}$ is the number of queries in the current task which contain the term t_i. We encode each query by its corresponding embedding vector representation V_q and take the cosine distance of these vectors while defining

d_{ij}. We consider a simple window decay $f(d) = 1[d < a]$ to only considers queries that are at most distance a from the current query for a given sub-task.

4.3.4 Coherence based Likelihood Function

In order to fully specify our model, we need to formulate the likelihood function which specifies the likelihood of observing a given collection of queries in a particular subtask partition. Given the collection of N queries pertaining to a search task, let $t(z)$ donates the subtask assignment, i.e., the set of queries assigned to a particular subtask. The likelihood function then factors into a weighted product of terms, each of which is the probability of the set of queries at each subtask. Let $jt(z)|$ be the number of subtasks and t_c^k be the set of indices that are assigned to subtask k. The likelihood term is:

$$p(q|t(z), G_0) = \prod_{k=1}^{|t(c)|} \psi(q_{t^k(z)}) p(q_{t^k(z)}|G_0) \tag{4.5}$$

$$= \prod_{k=1}^{|t(c)|} PMI_t exp^{\frac{-(PMI_t - b)^2}{c^2}} p(q_{t^k(z)}|G_0) \tag{4.6}$$

wherein, each of the subtask is weighted by its subtask coherence score $\psi(q_{t^k(z)})$ as defined in Section 4.3.1, while $p(q_{t^k(z)}|G_0)$ is the likelihood of each subtask, which in turn is described by the probability of observing queries in this subtask given the base distribution. We compute the marginal probability that the set of queries from each subtask are drawn independently from the same parameter, which itself is drawn from G_0. Each subtask term then becomes:

$$p(q_{t^k(z)}|G_0) = \int \Big(\prod_{i \in t^k(z)} p(q_i|\theta) \Big) p(\theta|G_0) d\theta \tag{4.7}$$

To maintain conjugacy and avoid additional layer of sampling, we model the query terms in a conjugate Dirichlet-Multinomial distribution wherein the integral is straightforward to compute. As can be seen from the likelihood equation, the contribution from each subtask is weighed by a task coherence score associated with the subtask. Having defined the subtask likelihood $p(q_{t^k(z)}|G_0)$ and the correspond-

ing subtask coherence function ($\psi(q_{t^k(z)})$), we now can fully specify the likelihood function of the entire subtask partitioning scheme, which we make use of to perform inference.

4.3.5 Posterior Inference

Posterior inference helps us determine the conditional distribution of the hidden variables given the query observations, which could then be used for exploratory analysis of the search tasks and how the different subtasks cluster, and is needed to compute the predictive distribution of a new query, given a set of observations. The posterior of the proposed *TE-coh-ddCRP* model is intractable to compute because the dd-CRP places a prior over a combinatorial number of possible customer configurations. We provide a general strategy for approximating the posterior using Monte Carlo Markov chain (MCMC) sampling. We aim to construct a Markov chain whose stationary distribution is the posterior of interest. For our *TE-coh-ddCRP* model, the state of the chain is defined by z_i, the query assignments for each query point. We will also consider $t(z)$, which are the subtask assignments that follow from the customer assignments. Let $\eta = \{D, \alpha, f, G_0\}$ denote the set of model hyperparameters. It contains the distances D, the scaling factor α, the decay function f , and the base measure G_0. Let q denote the query observations.

We employ Gibbs sampling wherein we iteratively draw from the conditional distribution of each latent variable given the other latent variables and observations. The Gibbs sampler iteratively draws from

$$p(q_i^{new}|q_{-i}, x) \propto p(q_i^{new})|D, \alpha) \\ p(x|z(c_{-i} \cup c_i^{new}), G_0) \tag{4.8}$$

The first term is the dd-CRP prior (4.2.4) and the second term is the likelihood of the observations under the subtask partition given by $t(z_{-i} \cup t_i^{(new)})$. This can be thought of as removing the current link from the i^{th} query and then considering how each alternative new link affects the likelihood of the observations.

Given that the likelihood term factorizes into a product of independent terms,

the Gibbs sampler need only compute terms that correspond to changes in the subtask partition. Consider the partition $t(z_{-i})$, which may have split a subtask, and the new partition $t(z_{-i} \cup t_i^{(new)})$. There are three cases to consider. First, z_i might link to itself - there will be no change to the likelihood function because a self-link cannot join two subtasks. Second, z_i might link to another subtask but cause no change in the partition. Finally, z_i might link to another subtask and join two subtasks k and l. The Gibbs sampler for the the proposed model thus becomes:

$$p(z_i^{(new)}|z_{-i}, q, \eta) \propto$$

$$\begin{cases} \alpha & \text{if } z_i^{(new)} \textit{is equal to } i \\ f(d_{ij}) & \text{if } z_i^{(new)} = j \textit{ does not} \\ & \quad \textit{join 2 tables} \\ f(d_{ij}) \frac{p(q_{t^k(z_{-i}) \cup t^l(z_{-i})}|G_0)}{p(q_{t^k(z_{-i})}|G_0) p(q_{t^l(z_{-i})}|G_0)} & \text{if } z_i^{(new)} = j \textit{ joins} \\ & \quad \textit{two tables} \end{cases}$$

The values $p(q_{t^k(z)}|G_0)$ for the different sets of subtasks partitions can be computed based on the the likelihood functions defined in Section 4.3.

The proposed Gibbs sampler enables us to perform inference on the proposed *TE-coh-ddCRP* model based on which we can extract the different subtasks embedded in query collections belonging to any complex search tasks. We next present our evaluation strategies which demonstrate the efficacy of the proposed subtask extraction method.

4.4 Experimental Evaluation

In this section, we evaluate the robustness of the proposed sub-task extraction framework. We perform a user judgement study to evaluate the quality of the extracted sub-tasks. Further, we perform quantitative experiments wherein we report how the compared approaches perform in terms of task coherence metrics. We

finally show how search engines could leverage insights about subtasks and help gauge user effort expended in search.

4.4.1 Dataset

We primarily use two different real world datasets for our experiments. We make use of the AOL log dataset (*Dataset 1*) which consists of 20M web queries collected over three months [163]. The AOL log is a very large and long-term collection consisting of about 20 million of Web queries issued by more than 657000 users over 3 months. The dataset comprises of five fields viz. the search query string, the query time stamp, the rank of the selected item (if any), the domain of the selected item's URL (if any), and a unique user identifier. In addition to the AOL search logs, we use backend search logs of users (*Dataset 2*) from a major US-based search engine for a period of 30 days from May 1, 2015 to May 31, 2015, and choose a random sample of over 2.6 million users where each user is identified by a unique IP address, with over 200 million search sessions. We filter out inactive users from our dataset who participate in $<$50 sessions, and focus instead on the more active user population.

4.4.2 Baselines

To compare the performance of the proposed subtask extraction algorithm, we baseline against a number of methods including state-of-the-art task extraction systems, in addition to parametric and non-parametric clustering approaches.

1. **QC-HTC/QC-WCC** [70]: frequently used search task identification methods. QC-WCC conducted clustering by dropping query-pairs with low weights, while QC-HTC considered the similarity between the first and last queries of two clusters in agglomerative clustering.

2. **BHTD** [20]: a recently proposed non-parametric bayesian method which extracts hierarchies of search tasks and subtasks. To make fair comparisons, we flatten out the hierarchy at the appropriate depth so as to obtain comparable number of task clusters.

3. **LDA Time-Window(TW)**: building on top of a standard LDA topic model [164], this model assumes queries belong to the same search task only if they lie in a fixed or flexible time window, and uses LDA to cluster queries into topics based on the query co-occurrences within the same time window. We tested time windows of various sizes and report results on the best performing window size.

4. **LDA Word-Related (LDA-WR)**: assumes that queries belongs to the same search task only if they share at least one query term, and uses LDA [164] to cluster queries into topics based on the co-occurrences of queries that share at least one term.

5. **CRP model** ([165]): the traditional Chinese Restaurant Process model which serves as a non-parametric clustering baseline.

Additionally, in order to gauge the contributions arising from the different aspects of the *TE-coh-ddCRP* model, we also consider a number of variants of the proposed approach:

1. **vanilla-ddCRP**: the vanilla version of the ddCRP with basic Dirichlet Multinomial (DCM) likelihood and cosine similarity between query terms as the distance measure.

2. **BNP Subtasks**: the Bayesian nonparametric subtask extraction algorithm without the coherence based likelihood objective.

3. **TE-coh-ddCRP**: the entire proposed model with task embedding based distances and coherence based likelihood function.

4.4.3 User Study

Owing to the absence of ground truth data on sub-task classification, we resort to user judgments in order to validate the quality of sub-tasks extracted.

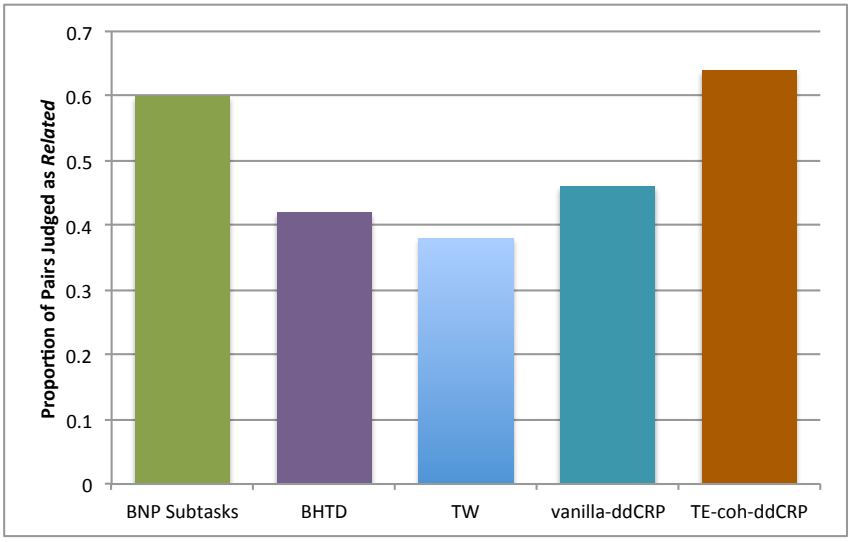


Figure 4.2: Judgments results for sub-task validity across compared approaches. The difference between TE-cor-ddCRP is statistically significant at p=0.05 level using a paired t-test.

4.4.3.1 Study Methodology

For the judgment study, we make use of the AOL search logs and sampled entire query history of frequent users who had more than 1000 unique search queries. We run the task extraction algorithms (as described in Section 7.2.1) on the entire set of queries of the sampled users to obtain a set of tasks from which we extract on-task queries to be used for further sub-task level analysis. We run the different baseline algorithms along with the proposed approach to these on-task queries to extract the sub-tasks. From among the set of subtasks extracted, we randomly selected a subset and collect judgments to access the quality of the sub-tasks extracted. Judgments were provided by judges who were recruited from the Amazon Mechanical Turk crowdsourcing service. Judges resided in the United States and were fluent in English. We restricted annotators to those based in the US because our logs came from searchers based in the US. We also used hidden quality control questions to filter out poor-quality judges.

4.4.3.2 Evaluating Subtask Coherence

In an ideal subtask extraction system, all the queries belonging to the same subtask cluster should ideally help a searcher solve the same information need, belong to the

same subtask and as a result have better subtask coherence. To this end, we evaluate the coherence property of the subtasks extracted by the different algorithms. We select a sub-task at random and then choose a randomly selected pair of queries from that sub-task. We then ask the human judges the following research question:

RQ: Subtask Relatedness: Are the given pair of queries related to the same sub-task? The possible judge options include (i) Related, (ii) Somewhat Related and (iii) Unrelated.

The subtask relatedness score provides an estimate of how coherent the extracted subtasks are. Indeed, a subtask cluster containing queries from different tasks would score less on Subtask Relatedness score since, if the cluster is impure, there is a greater chance that the 2 randomly picked queries would belong to different tasks and hence get judged as *Unrelated*. We repeat this process for a total of 100 iterations and compare the results with the ones obtained by our proposed approach, as well as with the ones obtained by few baselines. We report the proportion of query pairs judged as *Related* in Fig. 6.3. It is clear that our proposed method outperforms all the baselines considered, in making correct sub-task assignments.

4.4.4 Subtask Coherence Metric

In addition to the visual qualitative and judgement based evaluation, we also wish to measure coherence of the extracted subtasks. An ideal subtask would contain queries which all solve similar information needs. Recent work has demonstrated that it is possible to automatically measure topic coherence with near-human accuracy [166, 161] using a score based on pointwise mutual information (PMI). These studies show (using 6000 human evaluations) that the PMI-Score broadly agrees with human-judged topic coherence. We leverage the same insights and derive an estimate using PMI to capture subtask coherence. The Subtask Coherence measure indicates the atomicity of the information need associated with the subtask. It is captured by the semantic closeness of the queries associated with the subtask.

For each dataset, we treat two query terms as co-occurring if both terms occur

in the same search session. For a given subtask (t), we measure subtask coherence as the average of PMI scores for all pairs of the search terms associated with the task node:

$$Subtask\ Coherence\ = \frac{1}{|w|^2} \sum_{i=1}^{|w|} \sum_{j=1}^{|w|} PMI(w_i, w_j) \tag{4.9}$$

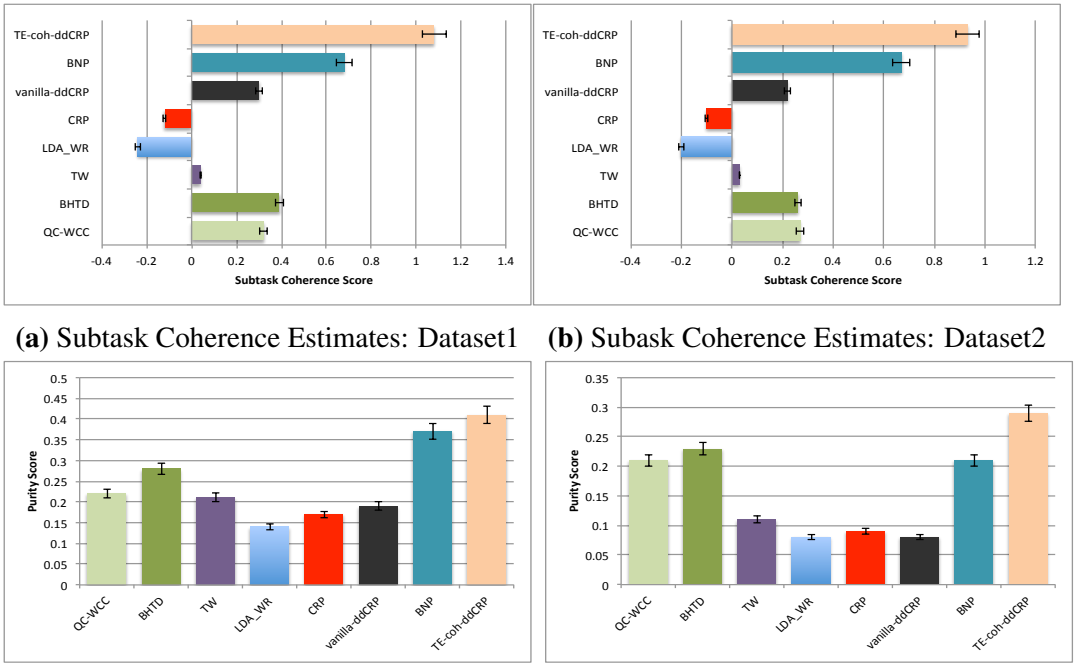
where $|w|$ represents the total number of unique search terms associated with subtask t. The subtask coherence score would be high for subtasks which contain queries that address similar information needs.

We present the results based on Subtask Coherence for the two datasets considered in Figures 4.3a & 4.3b. The proposed *TE-coh-ddCRP* model performs better than all baselines considered which shows that the tasks extracted are more coherent in terms of the information needs they address. The recently proposed task extraction systems QC-HTC and BHTD perform better than other baselines while LDA-TW approach performs the worst. By comparing the performance of coh-ddCRP and BNP, we are able to estimate the contributions by the different aspects of the proposed approach. Indeed, the coherence likelihood function helps improve performance over the vanilla-ddCRP model, while when combined with task based embeddings (TE-coh-ddCRP), the model outperforms all other methods in terms of task coherence.

Purity Estimates:

From a given collection of *on-task* queries, the proposed model simultaneously clusters queries into subtasks. Thus, the performance of identifying and labeling search subtasks mainly depends on how we cluster query words into different tasks. In the next few experiments, we evaluate the quality of obtained query clusters/subtasks, which in turn depends on their purity. Since no ground truth about the correct composition of a subtask is available, we assess purity by the average similarity of each pair of queries within the same subtask as:

$$Purity = \frac{1}{K} \sum_{k} \frac{\sum_{q_i, q_j \in t_k} sim(q_i, q_j)}{N_k(N_k - 1)/2} \times 100 \tag{4.10}$$



(a) Subtask Coherence Estimates: Dataset1 **(b)** Subask Coherence Estimates: Dataset2

(c) Purity Estimates: Dataset1 **(d)** Purity Estimates: Dataset2

Figure 4.3: Quantitative Evaluation of the subtasks extracted in terms of (i) Task Coherence and (ii) Purity estimates.

where N_k is the number of queries in subtask k. We evaluate the query similarity based on their task based vector embeddings as described in detail in Section 4.3.3.

Figure 4.3c and 4.3d compares the purity of topics detected by the proposed model, alternative probabilistic models, and state-of-the-art query clustering approaches on both the datasets considered. We observe that the proposed TE-coh-ddCRp model outperforms all compared approaches in terms of purity and is able to find subtask clusters wherein the query terms are more similar. It improves over the second best method by over 20%. We observe similar trends in terms of top performing baselines as we did while evaluating coherence. Among the variants of the proposed approach, the coherence enabled version performs better than the embedding enabled variant while their combination performs the best. This highlights the importance of considering both the embedding based distance metric aspect as well as the coherence based likelihood aspect, while extracting subtasks.

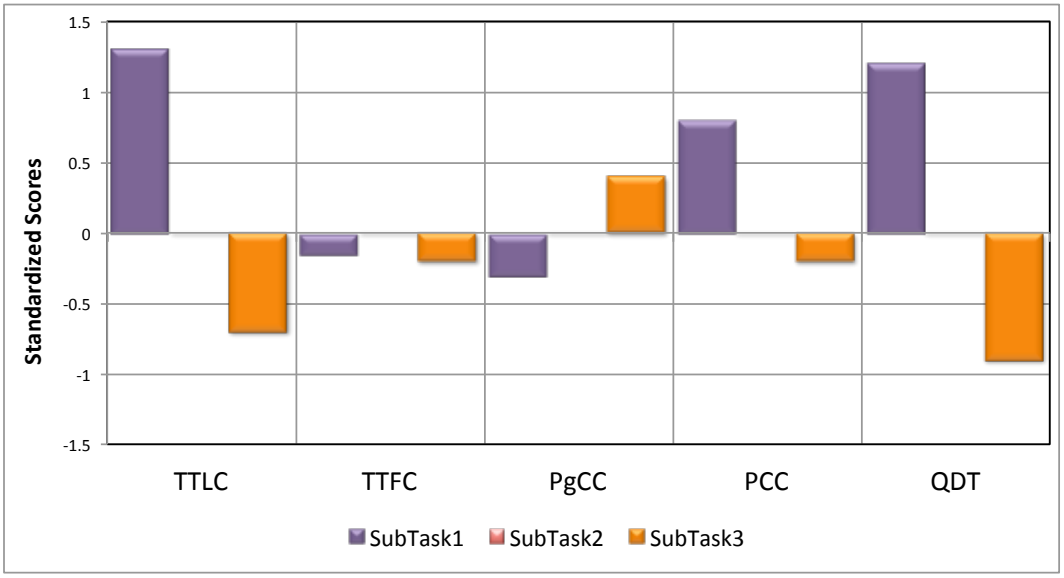


Figure 4.4: Effort metrics comparisons across different subtasks. To avoid publishing exact metric values, we treat Subtask 2 as 0 for normalization and report deviation scores for Subtask 1 and 3.

4.5 Subtask Efforts

Users invest significant time and effort in searching for information on search engines. Consequently, task effort metrics are an important indicator of user engagement with specific search tasks. In the current section, we make use of the proposed TE-coh-ddCRP model to extract real world subtasks and quantify some popular search metrics by aggregating across all queries for each of our identified sub-tasks. We highlight that task effort varies significantly across the different sub-tasks within a single task. Search engines need to be aware of such effort variations across subtasks in order to provide better task-aware personalizations.

4.5.1 Effort Metrics

We wish to understand the relationship between the different subtasks and total effort expended by the users in trying to accomplish them. We hypothesize that users expend different amounts of effort in completing the sub-tasks and this is reflected in the scores of task effort metrics that are recorded by search engines.

We analyse search effort across the different subtasks as extracted by our *TE-coh-ddCRP* model. We characterise search effort using 5 different metrics:

1. **Time to First Click (TTFC)**: measures the time elapsed before the user clicks the first link on the query result page. A longer TTFC is an indication of user surprise or confusion with the search results, and hints at a more extensive cognitive elaboration process while the user decides which link to click on.

2. **Time to Last Click (TTLC)**: measures the time elapsed before the user clicks the final link in her search session. The TTLC is a more direct measure of search effort expended by the user within a particular session. A higher TTLC could indicate user disatisfaction with the early results provided by the query results, or a heightened motivation on part of the user to search more about the particular topic of interest.

3. **Page Click Count (PCC)**: quantifies the total number of clicks observed on the search results page.

4. **Pagination Click Count (PgCC)**: quantifies the amount of pagination activity observed. Higher PgCC values indicate users examining results from beyond the first page. Both PCC and PgCC metrics are direct measures of the search effort put in by the user, and are hence good proxies to capture the different facets of search effort.

5. **Query Dwell Time (QDT):** quantifies the total time spent by the user on a particular query result before returning to the query result page or leaving the search engine. Thus, a higher QDT indicates a higher user interest and engagement with the selected query result and is therefore a good measure of the quality of query processing on part of the search engine.

4.5.2 Analysis

In this section, we provide an illustration of task effort differences within a single task. We choose ”plan wedding” task from the previous section and compute task effort metrics for the various subtasks, by aggregating these metrics over the constitutent queries for each of these three identified sub-tasks: **Sub-task 1:***Wedding*

dresses, **Sub-task 2:***Wedding hairstyles*, **Sub-task 3:***Wedding cards*.

Figure 4.4 illustrates that there exists significant differences in task effort metrics across the three sub-tasks. For instance, we find that wedding dresses (sub-task 1) and wedding hairstyles (sub-task 2) have a higher time to last click (TTLC) than wedding cards (sub-task 3). Conversely, however, the pagination click count (PgCC) for wedding cards is higher than both wedding hairstyles and wedding dresses. Understanding persistent differences in sub-task effort metrics is instrumental for search engines in developing context-aware personalizations.

4.6 Conclusion

Web search tasks are often complex and comprise several constituent sub-tasks. In this chapter, we offer a non-parametric Bayesian approach to identifying sub-tasks by grouping search queries using an embedding based dd-CRP approach. The proposed model combines insights from Bayesian nonparametrics and distributional semantics to extract subtasks which are not only meaningful but are also coherent. We evaluate our proposed method on a proprietary search logs dataset as well as on the AOL search logs and demonstrate the superiority over comparable approaches. We contend that our proposed approach is significantly more useful in online environments where the number of sub-tasks is never known a priori and impossible to ascertain or approximate. Further, using an embedding based distancing scheme we offer an improvement in empirical performance over prior clustering approaches that have used either a bag-of-words or TF-IDF based approach. Our method offers search engine providers with a novel method to identify and analyse user task-behavior, and better support task decisions on their platforms.

Chapter 5

Extracting Hierarchies of Search Tasks & Subtasks

5.1 Introduction

The need for search often arises from a person's need to achieve a goal, or a task such as booking travels, buying a house, etc., which would lead to search processes that are often lengthy, iterative, and are characterized by distinct stages and shifting goals [13]. Thus, identifying and representing these tasks properly is highly important for devising search systems that can help end users complete their tasks. It has previously been shown that these task representations can be used to provide users with better query suggestions [167], offer improved personalization [21, 168], provide better recommendations [169], help in satisfaction prediction [170] and search result re-ranking. Moreover, accurate representations of tasks could also be highly useful in aptly placing the user in the task-subtask space to contextually target the user in terms of better recommendations and advertisements, developing task specific ranking of documents, and developing task based evaluation metrics to model user satisfaction. Given the wide range of applications these tasks representations can be used for, significant amount of research has been devoted to task extraction and representation [73, 171, 6, 13, 76].

Task extraction is quite a challenging problem as search engines can be used to achieve very different tasks, and each task can be defined at different levels of gran-

ularity. A major limitation in existing task-extraction methods lies in their treatment of search tasks as flat structure-less clusters which inherently lack insights about the presence or demarcation of subtasks associated with individual search tasks. In reality, often search tasks tend to be hierarchical in nature. For example, a search task like planning a wedding involves subtasks like searching for dresses, browsing different hairstyles, looking for invitation card templates, finding planners, among others. Each of these subtasks (1) could themselves be composed of multiple subtasks, and (2) would warrant issuing different queries by users to accomplish them. Hence, in order to obtain more accurate representations of tasks, new methodologies for constructing hierarchies of tasks are needed.

As part of the proposed research, we consider the challenge of extracting hierarchies of search tasks and their associated subtasks from a search log given just the log data without the need of any manual annotation of any sort. We present an efficient Bayesian nonparametric model for discovering hierarchies and propose a tree based nonparametric model to discover this rich hierarchical structure of tasks/subtasks embedded in search logs. Most existing hierarchical clustering techniques result in binary tree structures with each node decomposed into two child nodes. Given that a complex task could be composed of an arbitrary number of subtasks, these techniques cannot directly be used to construct accurate representations of tasks. In contrast, our model is capable of identifying task structures that can be composed of an arbitrary number of children. We make use of a number of evaluation methodologies to evaluate the efficacy of the proposed task extraction methodology, including quantitative and qualitative analyses along with crowdsourced judgment studies specifically catered to evaluating the quality of the extracted task hierarchies. We contend that the techniques presented expand the scope for better recommendations and search personalization and opens up new avenues for recommendations specifically targeting users based on the tasks they involve in.

Symbol	Description
n_T	number of children of tree T
$ab\|c$	partition of set $\{a,b,c\}$ into disjoint sets $\{a,b\},\{c\}$
ch(T)	children of T
$\phi(T)$	partition of tree T
$p(D_m\|T_m)$	likelihood of data D_m given the tree T_m
π_{T_m}	mixing proportions of partition of tree T
$f(D_m)$	marginal probability of the data D_m
$\mathbb{H}(T)$	set of all partitions of queries $Q = leaves(T)$
$f(Q)$	task affinity function for set of queries Q
$r^k_{q_i,q_j}$	the k-th inter-query affinity between q_i & q_j

Table 5.1: Table of symbols

5.2 Defining Search Tasks

Jones et al. [13] was one of the first papers to point out the importance of task representations, where they defined a search task as:

definition A *search task* is an atomic information need resulting in one or more queries.

Follow-up work on tasks, specifically Lucchese *et al.et al.*[70] and Ahmed et al. [167] later extended this definition to a more generic one, which can also capture task structures that could possibly consist of related subtasks, each of which could be complex tasks themselves or may finally split down into simpler tasks or atomic informational needs. A complex search task can then be defined as [70, 167]:

definition A *complex search task* is a multi-aspect or a multi-step information need consisting of a set of related subtasks, each of which might recursively be complex.

The definition of complex tasks is much more generic, and captures all possible search tasks, that can be either complex or atomic (non-complex). Throughout this paper we adopt the definition provided in Definition 5.2.2 as the definition for a search task. Hence, by definition a search task has a hierarchical nature, where each task can consist of an arbitrary number of, possibly complex subtasks. An effective task extraction system should be capable of accurately identifying and representing such hierarchical structures.

5.3 Constructing Task Hierarchies

While hierarchical clustering are widely used for clustering, they construct binary trees which may not be the best model to describe data's intrinsic structure in many applications, for example, the task-subtask structure in our case. To remedy this, multi-branch trees are developed. Currently there are few algorithms which generate multi-branch hierarchies. Blundel *et al.* [145, 137] adopt a simple, deterministic, agglomerative approach called BRTs (Bayesian Rose Trees) for constructing multi-branch hierarchies. In this work, we adapt BRT as a basic algorithm and extend it for constructing task hierarchies. We next describe the major steps of BRT approach.

5.3.1 Bayesian Rose Trees

BRTs [145, 137] are based on a greedy probabilistic agglomerative approach to construct multi-branch hierarchies. In the beginning, each data point is regarded as a tree on its own: $T_i = \{x_i\}$ where x_i is the feature vector of i-th data. For each step, the algorithm selects two trees T_i and T_j and merges them into a new tree T_m. Unlike binary hierarchical clustering, BRT uses three possible merging operations, as shown in Figure 5.1:

- **Join**: $T_m = T_i, T_j$, such that the tree T_m has two children now
- **Absorb**: $T_m = children(T_i) \cup T_j$, i.e., the children of one tree gets absorbed into the other tree forming an absorbed tree with >2 children
- **Collapse**: $T_m = children(T_i) \cup children(T_j)$, all the children of both the subtrees get combined together at the same level.

Specifically, at each step, the algorithm greedily finds two trees T_i and T_j to merge which maximize the ratio of probability:

$$\frac{p(D_m|T_m)}{p(D_i|T_i)p(D_j|T_j)} \tag{5.1}$$

where $p(D_m|T_m)$ is the likelihood of data D_m given the tree T_m, D_m is all the leaf data of T_m, and $D_m = D_i \cup D_j$. The probability $p(D_m|T_m)$ is recursively defined on

the children of T_m:

$$p(D_m|T_m) = \pi_{T_m} f(D_m) + (1 - \pi_{T_m}) \prod_{T_i \in ch(T_m)} p(D_i|T_i) \tag{5.2}$$

where $f(D_m)$ is the marginal probability of the data D_m and π_{T_m} is the "*mixing proportion*". Intuitively, π_{T_m} is the prior probability that all the data in T_m is kept in one cluster instead of partitioned into sub-trees. In BRT[145], π_{T_m} is defined as:

$$\pi_{T_m} = 1 - (1 - \gamma)^{n_{T_m} - 1} \tag{5.3}$$

where n_{T_m} is the number of children of T_m, and $0 \geq \gamma \leq 1$ is the hyperparameter to control the model. A larger γ leads to coarser partitions and a smaller γ leads to finer partitions. Table 6.1 provides an overview of notations & symbols used throughout the paper.

5.3.2 Building Task Hierarchies

We next describe our task hierarchy construction approach built on top of Bayesian Rose Trees. A tree node in our setting is comprised of a group of queries which potentially compose a search task, i.e. these are the set of queries that people tend to issue in order to achieve the task represented in the tree node.

We define the task-subtask hierarchy recursively: T is a task if either T contains all the queries at its node (an atomic search task) or if T splits into children trees as $T = \{T_1, T_2, ..., T_{n_T}\}$ where each of the children trees (T_i) are disjoint set of queries corresponding to the n_T subtasks associated with task T. This allows us to consider trees as a nested collection of sets of queries defining our task-subtask hierarchical relation.

To form nested hierarchies, we first need to model the query data. This corresponds to defining the marginal distribution of the data $f(D_m)$ as defined in Equation 2. The marginal distribution of the query data ($f(D_m)$) helps us encapsulate insights about task level interdependencies among queries, which aid in constructing better task representations. The original BRT approach [145] assumes that the

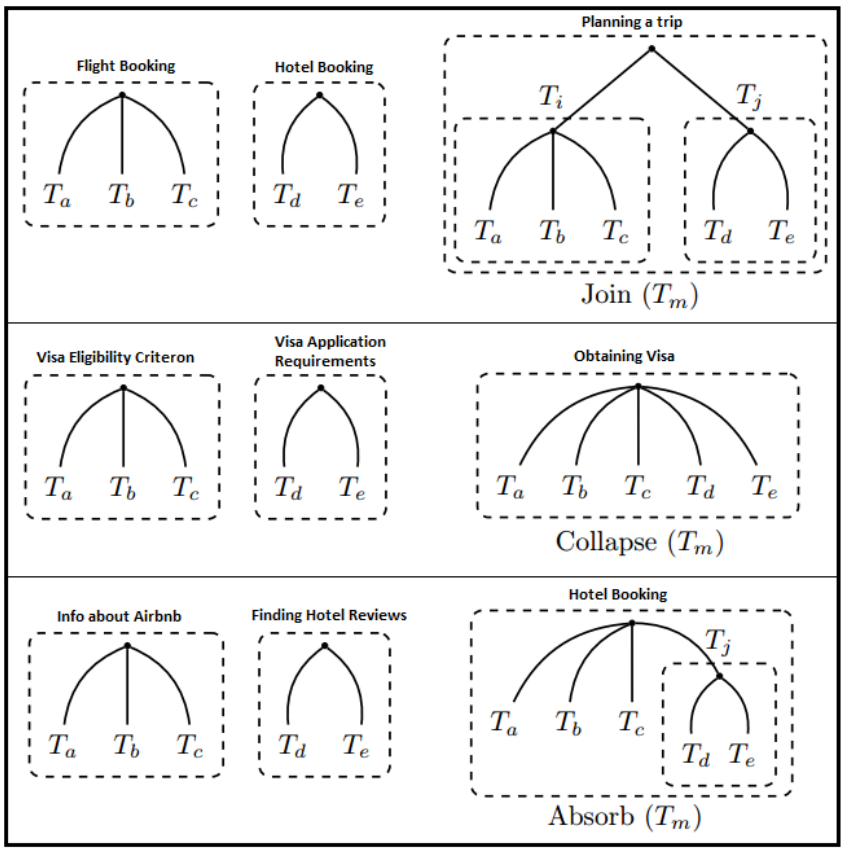


Figure 5.1: The different ways of merging trees which allows us to obtain tree structures which best explain the task-subtask structure.

data can be modeled by a set of binary features that follow the Bernoulli distribution. In other words, features (that represent the relationship/similarities between data points) are not weighted and can only be binary. Binary (0/1) relationships are too simplistic to model inter-query relationships; as a result, this major assumption fails to capture the semantic relationships between queries and is not suited for modeling query-task relations. To this end, we propose a novel query affinity model and to alleviate the binary feature assumption imposed by BRT, we propose a conjugate model of query affinities, which we describe next.

5.3.3 Conjugate Model of Query Affinities

A tree node in our setting is comprised of a group of queries which *potentially* belong to the same search task. The likelihood of a tree should encapsulate information about the different relationships which exists between queries. Our goal

Query-Term Based Affinity (r^1)	
cosine	cosine similarity between the term sets of the queries
edit	norm edit distance between query strings
Jac	Jaccard coeff between the term sets of the queries
Term	proportion of common terms between the queries
URL Based Affinity (r^2)	
Min-edit-U	Minimum edit distance between all URL pairs from the queries
Avg-edit-U	Average edit distance between all URL pairs from the queries
Jac-U-min	Minimum Jaccard coefficient between all URL pairs from the queries
Jac-U-avg	Average Jaccard coefficient between all URL pairs from the queries
Session/User Based Affinity (r^3)	
Same-U	if the two queries belong to the same user
Same-S	if the two queries belong to the same session
Embedding Based Affinity (r^4)	
Embedding	cosine distance between embedding vectors of the two queries

Table 5.2: Query-Query Affinities.

here is to make use of the rich information associated with queries and their result set available to compute the likelihood of a set of queries to belong to the same task. In order to do so, we propose a query affinity model which makes use of a number of different inter-query affinities to determine the tree likelihood function.

We next describe the technique used to compute four broad categories of inter-query affinity and later describe the Gamma-Poisson conjugate model which makes use of these affinities to compute the marginal distribution of the data.

Query-term based Affinity (r^1):

Search queries catering to the same or similar informational needs tend to have similar query terms. We make use of this insight and capture query level affinities between a pair of queries. We make use of cosine similarity between the query term sets, the normalized edit distances between queries and the Jaccard Coefficient between query term sets.

URL-based Affinity (r^2):

Users tackling similar tasks tend to issue queries (possibly different) which return similar URLs, thus encoding the URL level similarity between pairs of queries into

the query affinity model helps in capturing another task-specific similarity between queries. Any query pair having high URL level similarity increase the possibility of the query pair originating from similar informational needs. We capture a number of URL-based signals including minimum and average edit distances between URL domains and jaccard coefficient between URLs.

User/Session based Affinity (r^3):
It is often the case that users issue related queries within a session so as to satisfy their informational need. We leverage this insight by making use of session level information (as a 0/1 binary feature) and user-level information (as a 0/1 binary feature) in our affinity model to identify queries issued in the same session and by the same user accordingly.

Query Embedding based Affinity (r^4):
Word embeddings capture lexico-semantic regularities in language, such that words with similar syntactic and semantic properties are found to be close to each other in the embedding space. We leverage this insight and propose a query-query affinity metric based on such embeddings. We train a skip-gram word embeddings model where a query term is used as an input to a log-linear classifier with continuous projection layer and words within a certain window before and after the words are predicted. To obtain a query's vector representation, we average the vector representations of each of its query terms and compute the cosine similarity between two queries' vector representations to quantify the embedding based affinity (r^4).

Table 5.2 summarizes all features considered to compute these affinities. Our goal is to capture information from all four affinities when defining the likelihood of the tree. We assume that the global affinity among a group of queries can be decomposed into a product of independent terms, each of which represent one of the four affinities from the query-group. For each query group Q, we take the normalized sum of the affinities from all pairs of queries in the group Q to form each of the affinity component (r^k, k=1,2,3,4).

Poisson models have been shown as effective query generation models for information retrieval tasks [172]. While these affinities could be used with a lot of distributions, in the interest of computational efficiency and to avoid approximate solutions, our model will use a hierarchical Gamma-Poisson distribution to encode the query-query affinities. We incorporate the gamma-Poisson conjugate distribution in our model under the assumptions that the query affinities are discretized and for a group of queries Q, the affinities can be decomposed to a product of independent terms, each of which represents contributions from the four different affinity types. Finally, for a tree (T_m) consisting of the data (D_m), i.e. the set of queries Q, we define the marginal likelihood as:

$$f(D_m) = f(Q) = \prod_{k=1}^{k=4} p\Bigg(\sum_{i \in 1 \cdots |Q|} \sum_{j \in 1 \cdots |Q|} r^k_{q_i,q_j} | \alpha_k, \beta_k \Bigg) \tag{5.4}$$

where α_k & β_k are respectively the shape parameter & the rate parameter of the four different affinities. Making use of the Poisson-Gamma conjugacy, the probability term in the above product can be written as:

$$p(r|\alpha,\beta) = \int_\lambda p(r|\lambda) p(\lambda|\alpha,\beta) d\lambda \tag{5.5}$$

$$= \left\{ \frac{\Gamma(\alpha+r)}{r!\Gamma(\alpha)} \left(\frac{\beta}{\beta+1} \right)^{\alpha} \left(\frac{1}{\beta+1} \right)^{r} \right\} \tag{5.6}$$

where λ is the Poisson mean rate parameter which gets eliminated from computations because of the Gamma-Poisson conjugacy and where r, α & β get replaced by affinity class specific values.

5.3.4 Task Coherence based Pruning

The search task extraction algorithm described above provides us a way of constructing a task hierarchy wherein as we go down the tree, nodes comprising of complex multi-aspect tasks split up to provide finer tasks which ideally should model user's fine grained information needs. One key problem with the hierarchy construction algorithm is the continuous splitting of nodes which results in singleton queries

occupying the leave nodes. While splitting of nodes which represent complex tasks is important, the nodes representing simple search task queries corresponding to atomic informational needs should not be further split into children nodes. Our goal in this section is to provide a way of quantifying the task complexity of a particular node so as to prevent splitting up nodes representing atomic search task into further subsets of query nodes.

5.3.4.1 Identifying Atomic Tasks

We wish to identify nodes capturing search subtasks which represent atomic informational need. In order to do so, we introduce the notion of *Task Coherence*:

definition *Task Coherence* is a measure indicating the atomicity of the information need associated with the task. It is captured by the semantic closeness of the queries associated with the task.

By measuring Task Coherence, we intend to capture the semantic variability of queries within this task in an attempt to identify how complex or atomic a task is. For example, a tree node corresponding to a complex task like planning a vacation would involve queries from varied informational needs including flights, hotels, getaways, etc; while a tree node corresponding to a finer task representing an atomic informational need like finding discount coupons would involve less varied queries - all of which would be about discount coupons. Traditional research in topic modelling has looked into automatic evaluation of topic coherence [166] via Pointwise Mutual Information. We leverage the same insights to capture task coherence.

5.3.4.2 Pointwise Mutual Information

PMI has been studied variously in the context of collocation extraction [162] and is one measure of the statistical independence of observing two words in close proximity. We wish to compute PMI scores for each node of the tree. A tree node in our discussion so far has been represented by a collection of search queries. We split queries into terms and obtain a set of terms corresponding to each node, and calculate a node's PMI scores using the node's set of query terms.

For the PMI computation, we employ the AOL log query set and treat two query terms as co-occurring if both terms occur in the same session. For a given

task node, we measure task coherence as the average of PMI scores for all pairs of the search terms associated with the task node. The node's PMI-Score is used as the final measure of task coherence for the task represented via the corresponding node.

5.3.4.3 Tree Pruning

We use the task coherence score associated with each node of the task hierarchy constructed, and prune lower level nodes of the tree to avoid aggressive node splitting. The overall motivation here is to avoid splitting nodes which represent simple search tasks associated with atomic informational needs. We scan through all levels of the search task hierarchy obtained by the algorithm described above and for each node compute its task coherence score. If the task coherence score exceeds a specific threshold, it implies that all the queries in this particular node are aimed at solving the same or very similar informational need and hence, we prune off the sub-tree rooted at this particular node and ignore all further splits of this node.

5.3.5 Algorithmic Overview

We summarize the overall algorithm to construct the hierarchy by outlining the steps. The problem is treated as one of greedy model selection: each tree T is a different model, and we wish to find the model that best explains the search log data in terms of task-subtask structure.

Step 1: Forrest Initialization:
The tree is built in a bottom-up greedy agglomerative fashion, starting from a forest consisting of n (=$|Q|$) trivial trees, each corresponding to exactly one vertex. The algorithm maintains a forest F of trees, the likelihood $p(i) = p(D_i|T_i)$ of each tree $T_i \in F$ and the different query affinities. Each iteration then merges two of the trees in the forest. At each iteration, each vertex in the network is a leaf of exactly one tree in the forest. At each iteration a pair of trees in the forest F is chosen to be merged, resulting in forest $F*$.

Step 2: Merging Trees:

At each iteration, the best potential merge, say of trees X and Y resulting in tree I, is picked off the heap. Binary trees do not fit into representing search tasks since a task is likely to be composed of more than two subtasks. As a result, following [137] we consider three possible mergers of two trees T_i and T_j into T_m. T_m may be formed by joining T_i and T_j together using a new node, giving $T_m = \{T_i, T_j\}$. Alternatively T_m may be formed by absorbing T_i as a child of T_j, yielding $T_m = \{T_j\} \bigcup ch(T_i)$, or vice-versa, $T_m = \{T_i\} \bigcup ch(T_j)$. We explain the different possible merge operations in Figure 5.1. We obtain arbitrary shaped sub-trees (without restricting to binary tress) which are better at representing the varied task-subtask structures as observed in search logs with the structures themselves learnt from log data. Such expressive nature of our approach differentiates it from traditional agglomerative clustering approaches which necessarily result in binary trees.

Step 3: Model Selection:

Which pair of trees to merge, and how to merge these trees, is determined by considering which pair and type of merger yields the largest Bayes factor improvement over the current model. If the trees T_i and T_j are merged to form the tree M, then the Bayes factor score is:

$$SCORE(M;I,J) = \frac{p(D_M|F*)}{p(D_M|F)} \tag{5.7}$$

$$= \frac{p(D_M|M)}{p(D_i|T_i)p(D_j|T_j)} \tag{5.8}$$

where $p(D_i|T_i)$ and $p(D_j|T_j)$ are given by the dynamic programming equation mentioned above. After a successful merge, the statistics associated with the new tree are updated. Finally, potential mergers of the new tree with other trees in the forest are considered and added onto the heap.

The algorithm finishes when no further merging results in improvement in the Bayes Factor score. Note that the Bayes factor score is based on data local to the merge - i.e., by considering the probability of the connectivity data only among

the leaves of the newly merged tree. This permits efficient local computations and makes the assumption that local community structure should depend only on the local connectivity structure.

Step 4: Tree Pruning:
After constructing the entire hierarchy, we perform the post-hoc tree pruning procedure described in Section 5.3.4 wherein we identify atomic task nodes via their task coherence estimates and prune all child nodes of the identified atomic nodes.

5.4 Experimental Evaluation

We perform a number of experiments to evaluate the proposed task-subtask extraction method. First, we compare its performance with existing state-of-the-art task extraction systems on a manually labelled ground-truth dataset and report superior performance (5.4.1). Second, we perform a detailed crowd-sourced evaluation of extracted tasks and additionally validate the hierarchy using human labeled judgments (5.4.2). Third, we show a direct application of the extracted tasks by using the task hierarchy constructed for term prediction (5.4.3).

Parameter Setting:
Unless stated otherwise, we made use of the best performing hyperparameters for the baselines as reported by the authors. The query affinities in the proposed approach were computed from the specific query collection used in the dataset used for each of the three experiments reported below. While hyperparmeter optimization is beyond the scope of this work, we experimented with a range of the shape and inverse scale hyperparameters (α, β) used for the Poison Gamma conjugate model and used the ones which performed best on the validation set for the search task identification results reported in the next section. Additionally, for the tree pruning threshold, we empirically found that a threshold of 0.8 gave the best performance on our toy hierarchies, and was used for all future experiments.

5.4.1 Search Task Identification

To justify the effectiveness of the proposed model in identifying search tasks in query logs, we employ a commonly used AOL data subset with search tasks annotated which is a standard test dataset for evaluating task extraction systems. We used the task extraction dataset as provided by Lucchese *et al.*[70]. The dataset comprises of a sample of 1000 user sessions for which human assessors were asked to manually identify the optimal task-based query sessions, thus producing a ground-truth that can be used for evaluating automatic task-based session discovery methods. For further details on the dataset and the dataset access links, readers are directed to Lucchese *et al.*[70].

We compare our performance with a number of search task identification approaches:

- **Bestlink-SVM** [10]: This method identified search task using a semi-supervised clustering model based on the latent structural SVM framework.

- **QC-HTC/QC-WCC** [70]: This series of methods viewed search task identification as the problem of best approximating the manually annotated tasks, and proposed both clustering and heuristic algorithms to solve the problem.

- **LDA-Hawkes** [76]: a probabilistic method for identifying and labeling search tasks that model query temporal patterns using a special class of point process called Hawkes processes, and combine topic model with Hawkes processes for simultaneously identifying and labeling search tasks.

- **LDA Time-Window(TW)**: This model assumes queries belong to the same search task only if they lie in a fixed or flexible time window, and uses LDA to cluster queries into topics based on the query co-occurrences within the same time window. We tested time windows of various sizes and report results on the best performing window size.

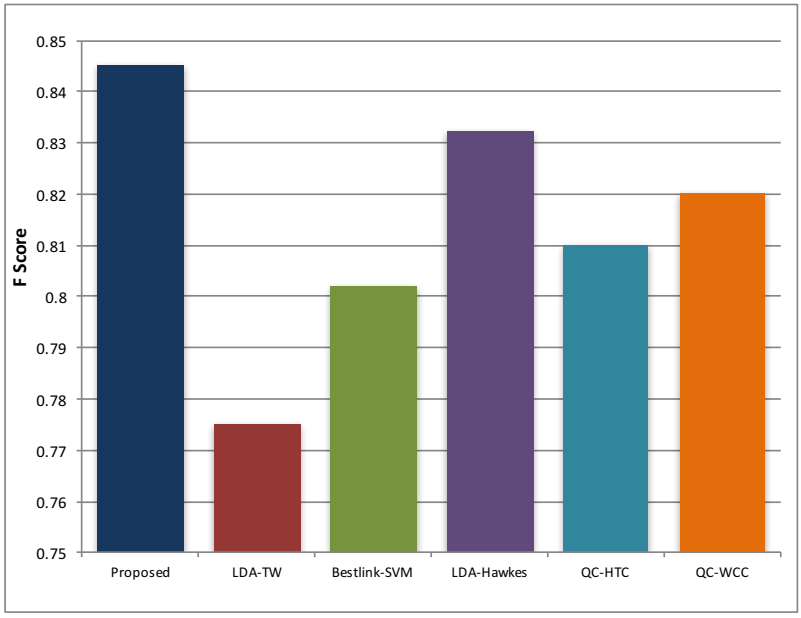


Figure 5.2: F1 score results on AOL tagged dataset

5.4.1.1 Metrics

A commonly used evaluation metric for search task extraction is the pairwise F-measure computed based on pairwise precision/recall [13, 6] defined as,

$$p_{pair} = \frac{\Sigma_{i \leq j} \delta(y(q_i), y(q_j)) \delta(\hat{y}(q_i), \hat{y}(q_j))}{\delta(\hat{y}(q_i), \hat{y}(q_j))} \tag{5.9}$$

$$r_{pair} = \frac{\Sigma_{i \leq j} \delta(y(q_i), y(q_j)) \delta(\hat{y}(q_i), \hat{y}(q_j))}{\delta(y(q_i), y(q_j))} \tag{5.10}$$

where p_{pair} evaluates how many pairs of queries predicted in the same task, i.e., $\delta(\hat{y}(q_i), \hat{y}(q_j) = 1$, are actually annotated as in the same task, i.e., $\delta(y(q_i), y(q_j)) = 1$ and r_{pair} evaluates how many pairs annotated as in the same task are recovered by the algorithm. Thus, globally F-measure evaluates the extent to which a task contains only queries of a particular annotated task and all queries of that task. Given p_{pair} and r_{pair}, the F-measure is computed as:$F_1 = \frac{2 \times p_{pair} \times r_{pair}}{p_{pair} + r_{pair}}$.

5.4.1.2 Results & Discussion

Figure 5.2 compares the proposed model with alternative probabilistic models and state-of-the-art task identification approaches by F1 score. To make fair comparisons, we consider the last level of the pruned tree constructed as task clusters when

computing pairwise precision/recall values. It is important to note that the labelled dataset has only flat tasks extracted on a per user basis; as a result, this dataset is not ideal for making fair comparisons of the proposed hierarchy extraction method with baselines. Nevertheless, the proposed approach manages to outperform existing task extraction baselines while having much greater expressive powers and providing the subdivision of tasks into subtasks. LDA-TW performs the worst since its assumptions on query relationship within the same search task are too strong. The advantage over QC-HTC and QC-WCC demonstrates that appropriate usage of query affinity information can even better reflect the semantic relationship between queries, rather than exploiting it in some collaborative knowledge.

5.4.2 Evaluating the Hierarchy

While there are no gold standard datasets for evaluating hierarchies of tasks, we performed crowd-sourced assessments to assess the performance of our hierarchy extraction method. We separately evaluated the coherence and quality of the extracted hierarchies via two different set of judgements obtained via crowdsourcing.

Evaluation Setup

For the judgment study, we make use of the AOL search logs and randomly sampled entire query history of frequent users who had more than 1000 search queries. The AOL log is a very large and long-term collection consisting of about 20 million of Web queries issued by more than 657000 users over 3 months. We run the task extraction algorithms on the entire set of queries of the sampled users and collect judgments to assess the quality of the tasks extracted. Judgments were provided by over 40 judges who were recruited from the Amazon Mechanical Turk crowdsourcing service. We restricted annotators to those based in the US because the logs came from searchers based in the US. We also used hidden quality control questions to filter out poor-quality judges. The judges were provided with detailed guidelines describing the notion of search tasks and subtasks and were provided with several examples to help them better understand the judgement task.

Evaluating Task Coherence

In the first study, we evaluated the quality of the tasks extracted by the task extraction algorithms. In an ideal task extraction system, all the queries belonging to the same task cluster should ideally belong to the same task and hence have better task coherence. To this end, we evaluate the task coherence property of the tasks extracted by the different algorithms. For each of the baselines and the proposed algorithm, we select a task at random from the set of tasks extracted and randomly pick up two queries from the selected task. We then ask the human judges the following question:

RQ1: Task Relatedness: Are the given pairs of queries related to the same task? The possible options include (i) Task Related, (ii) Somewhat Task Related and (iii) Unrelated.

The task relatedness score provides an estimate of how coherent tasks are. Indeed, a task cluster containing queries from different tasks would score less in Task Relatedness score since if the task cluster is impure, there is a greater chance that the 2 randomly picked queries belong to different tasks and hence get judged Unrelated.

Evaluating the hierarchy

While there are no gold standard dataset to evaluate hierarchies, in our second crowd-sourced judgment study, we evaluate the quality of the hierarchy extracted. A valid task-subtask hierarchy would have the parent task representing a higher level task with its children tasks representing more focused subtasks, each of which help the user achieve the overall task identified by the parent task.

We evaluate the correctness of the hierarchy by validating parent-child task-subtask relationships. More specifically, we randomly select a parent node from the hierarchy and then randomly select a child node from the set of its immediate child nodes. Given such parent-child node pairs, we randomly pick 5 queries from the parent node and randomly pick 2 queries from the child node. We then show the

human judges these parent and child queries and ask the following questions:

RQ2: Subtask Validity: Consider the set of queries representing the search task and the pair of queries representing the subtask. How valid is this subtask given the overall task?

The possible judge options include (i) Valid Subtask, (ii) Somewhat valid and (iii) Invalid. Answering this question helps us in analyzing the correctness of the parent-child task-subtask pairs.

RQ3: Subtask Usefulness: Consider the set of queries representing the search task and the pair of queries representing the subtask. Is the subtask useful in completing the overall search task?

The possible judge options include (i) Useful, (ii) Somewhat Useful and (iii) Not Useful. This helps us in evaluating the usefulness of task-subtask pairs by finding the proportion of subtasks which help users in completing the overall task described by the parent node. Overall, the RQ2 and RQ3 help in evaluating the correctness and usefulness of the hierarchy extracted.

Baselines

Since RQ1 evaluates task coherence without any notion of task-subtask structure, we compare against the top performing baselines from the task extraction setup described in section 5.4.1. On the other hand, RQ2 & RQ3 help in answering questions about the quality of hierarchy constructed. To make fair comparisons while evaluating the hierarchies, we introduce additional hierarchy extraction baselines:

- **Jones Hierarchies** [13]: A supervised learning approach for task boundary detection and same task identification. We train the classifier using the supervised Lucchese AOL task dataset and use it to extract tasks on the current dataset used in the judgment study.
- **BHCD** [137]: A state-of-the-art bayesian hierarchical community detection

	Task Relatedness				
	Proposed	**LDA-TW**	**QC-WCC**	**LDA-Hawkes**	**QC-HTC**
Task Related	**72%***	47%	60%	67%	61%
Somewhat Related	20%	14%	15%	13%	5%
Unrelated	**10%**	23%	25%	20%	34%

Table 5.3: Performance on Task Relatedness. The results highlighted with * signify statistically significant difference between the proposed approach and best performing baseline using χ^2 test with $p \leq 0.05$.

	Subtask Validity			
	Proposed	**Jones**	**BHCD**	**BAC**
Valid	**81%***	69%	51%	49%
Somewhat Valid	8%	19%	17%	21%
Not Valid	11%	12%	32%	30%
	Subtask Usefulness			
Useful	**67%***	52%	41%	43%
Somewhat Useful	8%	17%	19%	20%
Not Useful	25%	31%	40%	37%

Table 5.4: Performance on Subtask Validity and Subtask Usefulness. Results highlighted with * signify statistically significant difference between the proposed framework and best performing baseline using χ^2 test with $p \leq 0.05$.

algorithm based on stochastic blockmodels and makes use of Beta-Bernoulli conjugate priors to define a network. We build a network of queries and apply BHCD algorithm to extract hierarchies of query communities.

- **Bayesian Agglomerative Clustering (BAC)** [139]: A standard agglomerative hierarchical clustering model based on Dirichlet process mixtures.

Results & Discussion

For the first judgment study, each HIT is composed of 20 query pairs per approach being judged for task relatedness. We had three judges work on every HIT. Overall, per method we obtained judgments for 60 query pairs to evaluate the performance on task-relatedness. From among the three judges judging each query-pair, we followed majority voting mechanism to finalize the label for the instance. Table 5.3 presents the proportions of query pairs judged as related. About 72% of query pairs were judged task-related for the proposed approach with LDA-Hawkes performing second best with 67%. Task relatedness measures how pure the task clusters ob-

tained are, a higher score indicates that the queries belonging to the same task are indeed used for solving the same search task. The overall results indicate that the tasks extracted by the proposed task-subtask extraction algorithm are indeed better than those extracted by the baselines.

For the second judgment study used for evaluating the quality of the hierarchy, we show 10 pairs of parent-child questions in each HIT and ask the human annotators to judge the subtask validity and usefulness. Overall, per method we evaluate 300 such judgments resulting in over 1200 judgments and used maximum voting criterion from among the 3 judges to decide the final label for each instance. Table 5.4 compares the performance of the proposed hierarchy extraction method against other hierarchical baselines. The identified subtask was found useful in 67% cases with the best performing baseline being useful in 52% of judged instances. This highlights that the extracted hierarchy is indeed composed of better subtasks which are found to be useful in completing the overall task depicted by the parent task. It is interesting to note that for BHCD and BAC baselines, most often the subtasks were found to be invalid and not useful.

Since the same parent-child task-subtask was judged for validity and usefulness, it is expected that the proportion of task-subtasks judged useful would be less than the ones judged valid. Indeed, as can be seen from the Table 5.4, the relative proportions of tasks-subtasks found useful is much less than those found valid.

5.4.3 Term Prediction

In addition to task extraction and user study based evaluation, we chose to follow an indirect evaluation approach based on *Query Term Prediction* wherein given an initial set of queries, we predict future query terms the user may issue later in the session.This is in line with our goal of supporting users tackling complex search tasks since a task identification system which is capable of identifying "*good*" search tasks will indeed perform better in predicting the set of future query terms.

To evaluate the performance of the proposed task extraction method, we primarily work with the TREC Session Track 2014 [173] and AOL log data and constructed a new dataset consisting of user sessions from AOL logs concerned with

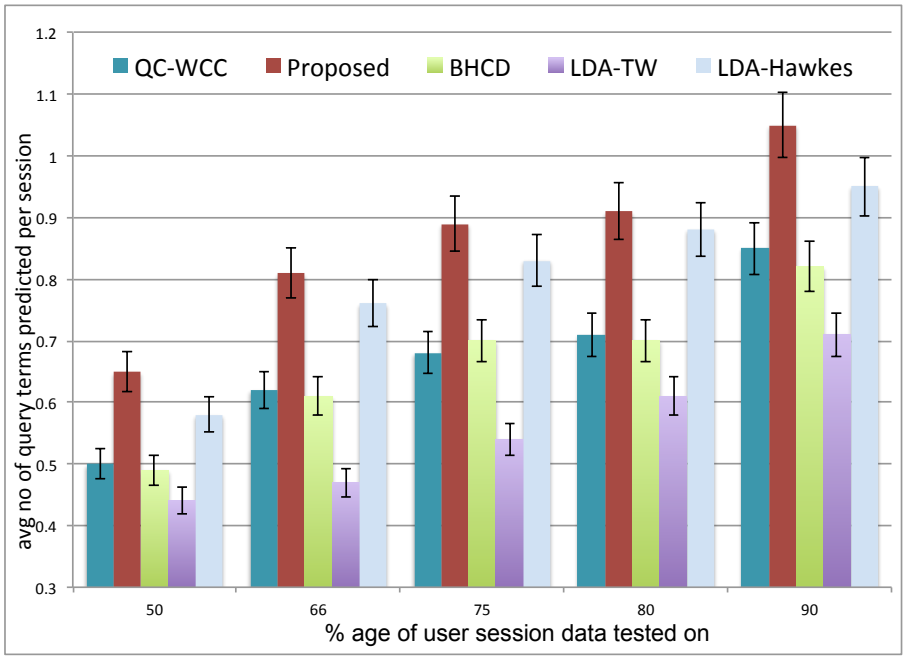


Figure 5.3: Term Prediction performance

Session Track queries. The session track data consists of over 1200 sessions while AOL logs consists of 20M search queries issued by over 657K users. We find the intersection of queries between the Session Track data and AOL logs to identify user sessions in AOL data trying to achieve similar task objectives. The Session Track dataset consists of 60 different ***topics***. For each of these 60 topics, we separately find user sessions from the entire AOL logs which contain query overlaps with these topics. For each topic, we iterate through the entire AOL logs and select any user session which contains query overlap with the current topic. As a result, we obtain a total of 14030 user sessions which contain around 6.4M queries.

Given the initial queries from a user session and a set of tasks extracted from Session Track data, we leverage queries from the identified task to predict future query terms. For each Session Track topic, we construct a task hierarchy and use the constructed task hierarchy to predict future query terms in the associated user sessions. More specifically, for each topic, we split each user session into two parts: (i) task matching and (ii) held-out evaluation part. We use queries from the task matching part of user sessions to obtain the right node in the task hierarchy from which we then recommend query terms. We pick the tree node which has the

highest cosine similarity score based on all the query terms under consideration. We evaluate based on the absolute recall scores - the average number of recommended query terms which match with the query terms in the held-out evaluation part of user sessions.

We baseline against the top performing task extraction baselines from Section 5.4.1 as well as the top performing hierarchical algorithms from Section 5.4.2. To make fair comparisons, we consider nodes at the bottom most level of the pruned tree for task matching and term recommendation.

Figure 5.3 compares the performance on term prediction against the considered baselines. We plot the average number of query terms predicted against the proportion of user session data used. The proposed method is able to better predict future query terms than a standard task extraction baseline as well as a very recent hierarchy construction algorithm.

5.5 Conclusion

Search task hierarchies provide us with a more naturalistic view of considering complex tasks and representing the embedded task-subtask relationships. In this chapter, we first motivated the need for considering hierarchies of search tasks & subtasks and presented a novel bayesian nonparametric approach which extracts such hierarchies. We introduced a conjugate query affinity model to capture query affinities to help in task extraction. Finally, we propose the idea of Task Coherence and use it to identify atomic tasks. Our experiments demonstrated the benefits of considering search task hierarchies. Importantly, we were able to demonstrate competitive performance while at the same time outputting a richer and more expressive model of search tasks. This expands the scope for better task recommendation, better search personalization and opens up new avenues for recommendations specifically targeting users based on the tasks they are involved in.

Part II

Leveraging Task Information

Chapter 6

Terms, Topics & Tasks: Enhanced User Modelling for Better Personalization

6.1 Introduction

As consumers of the informational content, different users have distinct information seeking preferences; thus accurately understanding their respective information needs and decision preferences is crucial for providing effective support during search interactions. While user behaviours are largely determined by their own goals and preferences, the mined knowledge from log activity data reveals different user intentions and behaviour patterns, which provide unique signals for user centric optimization and personalization.

Web search personalization has recently received a lot of attention by the research community. Personalized search leverages information about an individual to identify the most relevant recommendations for that person. A challenge for personalization is in collecting user profiles that are sufficiently rich to be useful in settings such as result ranking and query recommendations.

Most previous work on personalization has focused on using long term search histories to provide better personalized results. In particular, most recent personalized search systems mainly focus on identifying topics a user might be interested in

based on their search history and improving their search experience by identifying and using information from different topics [95, 93].

Even though using topical interest of users can be highly valuable in personalizing search results and improving user experience, it still ignores the fact that two different users that have similar topical interests may still be interested in achieving very different tasks with respect to this topic. For example, a stockbroker and a normal investor while being interested in the same topic (finance), perform quite different set of search tasks and as a result need different kinds and levels of support while tackling these tasks. More generally, while topical interests capture the heterogeneity among users stemming from varied topical interests, such task based approaches would assist in capturing the heterogeneity stemming from differences in user needs and behaviors. Hence, using task information together with topics could result in systems that can provide improved personalized search experience to users.

In this chapter, we focus on using *search task* information for user modeling, where a search task has been previously defined as an atomic information need that consists of a set of related (sub)tasks [174]. In a recent poster [175], we showed that search tasks can indeed be used for personalization. This work was based on replacing topic models with search tasks for personalization and building task based representations of users for topic modelling. Hence, this work ignores the fact that tasks users are interested in tend to be topic specific: people tend to be interested in achieving certain tasks only for certain topics. In this work, we investigate the idea of task based personalization in detail and develop a model that combines topic based user modelling with task based user models. Additionally, we look at the user's search history that provides information about user's term usage behavior. We integrate user's historical information to the task-topic tensor framework by proposing a coupled matrix-tensor factorization model which jointly learns user representations based on their search history, term usage behavior, topical interest profiles and search task behaviors.

In particular, we show that it is possible to represent the topic specific tasks

users are interested in by representing users in terms of a 3-modal $< user - topic - task >$ tensor (multidimentional array). We show that tensor factorization can be used to learn *coupled task-topic based user representations* for each user, thereby incorporating tasks together with topics in representing the user population. The tensor based framework helps in encapsulating the complex interactions between topics and tasks across the entire user population and learns a low dimensional factor model wherein user's interests, preferences and behaviors are determined by an interplay between these latent factors. We further extend the tensor based framework to include user's search history information by proposing the use of coupled matrix-tensor factorization model [176] wherein the matrix captures user's topical interest and search task information while the matrix captures user's term usage behavior.

Finally, we show that the proposed methods result in better user profiles by evaluating the quality of our approach on a variety of tasks for personalisation including collaborative query recommendation, cluster based recommendation and user cohort analysis.

6.2 Methodology

We propose a new direction in learning user representations by modeling user's task behaviors. We posit that topics and tasks capture different set of insights about user's behavior and information needs and can be coupled with their term usage behavior to jointly learn richer user representations, which is the main goal of this work.

To this end, we intend to extract search tasks from a given search log and represent users in terms of these tasks. In the next sub-section, we describe the approach we use to extract search tasks. This is followed by briefly describing our initial efforts in modeling users based on tasks alone ignoring the topical information [175] in section 6.3. Finally, we present our approach of coupling task and topical information in Section 6.4 and extend it to include user's language model and term usage behavior in Section 6.5. We describe the experimental evaluation set up and results

Symbol	Description
ALS	Alternating Least Squares
CMTF	Coupled Matrix Tensor Factorization
$\mathbf{A} \odot \mathbf{B}$	Khatri-Rao product
$a \circ b \circ c$	$(a \circ b \circ c)(i,j,k) = a(i)b(j)c(k)$
$\mathbf{A}_1^i$	series of matrices or vectors, indexed by i
$\|A\|_F$	Frobenius norm of $\mathbf{A}$
$\mathbf{T}$	User-Topic-Task tensor
$\mathbf{M}$	User-Term matrix
$\mathbf{U}$	User representation matrix
$\mathbf{S}$	Search Task matrix
$\mathbf{L}$	LDA topics matrix
$\mathbf{W}$	User language model matrix

Table 6.1: Table of symbols

in Section 6.6, while section 6.7 concludes.

6.2.1 Notation & Background

We start with defining the notations used throughout the chapter. Columns of a matrix are denoted by boldface lower letters with a subscript, e.g., $\mathbf{a_r}$ is the $r-th$ column of matrix $\mathbf{A}$. Entries of a matrix or a tensor are denoted by lowercase letters with subscripts, i.e., i_1 entry. Given two matrices $\mathbf{A} \in \Re^{I \times K}$ and $\mathbf{B} \in \Re^{J \times K}$, their Khatri-Rao product is denoted by $A \odot B$ and defined as column-wise Kronecker product. The result is a matrix of size $(IJ) \times K$ and defined by

$$A \odot B = [a_1 \otimes b_1 a_2 \otimes b_2 \ldots a_K \otimes b_K] \tag{6.1}$$

where $\otimes$ denotes Kronecker product. For more details on properties of Kronecker and Khatri-Rao products, the reader is referred to *Kolda et al.* [149].

Table 6.1 shows a list of symbols used throughout the chapter, together with their descriptions.

6.2.2 Extracting Search Tasks

In order to build task based representaions of users, we first need to identify and extract search tasks users are likely to perform when they use a search engine. Here we describe our approach of extracting these tasks given a search log. Following

the approach in Lucchese *et al.* [70], we employ a graph based query-clustering approach based on finding weighted connected components of a graph.

Given a user session ϕ, we build a complete graph $G_\phi = (V, E, w)$, whose nodes V are the queries in ϕ, and whose E edges are weighted by the similarity of the corresponding nodes. The weighting function w is a similarity function $w : E \rightarrow R \in [0,1]$ that can be easily instantiated in terms of the distance functions μ, which we describe a bit later. The graph G_ϕ describes the similarity between any pair of queries in the given session. For evaluating similarity between two queries, we make use of the following two similarity features:

- **Content-based**: Two queries that share some common terms are likely related. Sometimes, such terms may be very similar, but not identical, due to mispelling, or different prefixes/suffixes. To capture content distance between queries, following Lucchese *et al.* [70] we adopt a Jaccard index on tri-grams along with a normalized Levenstein distance which is widely accepted as the best edit-based feature for identifying goal boundaries [70].

- **Semantic-based**: Following Lucchese *et al.* [70], we assume that a Wikipedia article describes a certain concept and that the presence of a term in a given article is an evidence of the correlation between that term and that concept. We represent each term in a high-dimensional concept space, and sum over each query term to obtain a query's concept vectors. The cosine similarity between such concept-vectors of queries provides the semantic similarity between the two queries. The distance between two queries is defined as a (1- weighted average of the two similarities). For further details, users are referred to Lucchese *et al.* [70].

Based on the query pair distances obtained above, weak edges with low similarity are dropped, since the corresponding queries are not related, and clusters are built on the basis of the strong edges, i.e. with high similarity, which identify the related query pairs. The connected components of the pruned query-query graph identify the clusters of related queries and provides us with our set of search tasks.

Lucchese *et al.* [70] provide further details on the above mentioned similarity features.

6.3 Learning Task Based User Representations

We postulate that in a web search setting, search logs contain information about various actions that users perform and profiling users based on search tasks would better capture the heterogeneity in user information and help us in modeling users. In a recent poster [175], we present some preliminary work which describes a purely search task based user representation system (ignoring topical information) as described in this section. We later propose a novel way of combining such task based representations with user's topical interest information to learn a coupled task-topical interest user profile and additionally incorporate user's term histories via a coupled matrix-tensor factorization framework described in Section 6.5.

User-Task Association Matrix: Based on the extracted search tasks, we construct a user-task association matrix which represents the search tasks users have been involved with. For each user u_i, we consider their search history and create a bag-of-queries representation from the list of queries issued by the user and compare each user with each of the search tasks t_j obtained by the method described in section 6.2.2. For each user-task $< u_i, t_j >$ pair, we populate the corresponding value in the user-task association matrix (R) with the cosine similarity score (r_{ij}) we obtain for the pair. For tasks in which users do not have any matching queries, we assign a score of 0 to the corresponding pair. The overall motivation behind such a set-up is to capture information about whether or not users have performed such a search task before.

Probabilistic Matrix Factorization for User Representations: We wish to extract task-based user vector representations by jointly mapping users and tasks to a joint latent factor space. Following Salakhutdinov *et al.* [177], we model the user-task association in terms of probabilistic matrix factorization problem and

learn latent vector representation for each user from the user-task association matrix by fitting a probabilistic model. Given the user-task association matrix **R**, we find the user feature matrix **U** = $[u_i]$ and task feature matrix **T** = $[t_j]$. The conditional distribution over the observed user-task associations $R \in \Re^{N \times M}$ is given by:

$$P(R|U,T,\alpha) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T T_j, \sigma^2)\right]^{I_{ij}} \tag{6.2}$$

where $\mathcal{N}$ denotes the Gaussian distribution and I_{ij} is the indicator function which is 1 if the user i was involved in search task j. The latent vector representations for the users, the system minimizes the regularized error:

$$min_{u^*,t^* \in \kappa} \sum_{i,j} (r_{ij} - t_j^T u_i)^2 + \lambda\left(\|u_i\|^2 + \|t_j\|^2\right) \tag{6.3}$$

where κ is the set of non-zero r_{ij} values, u_i represents the user and t_j represents a task. The user matrix **U** obtained as a result, contains vector representations of each of the users which is used in further experiments.

So far, we have been able to extract collective search tasks from all users and learnt a user representation based on these search tasks. We show in Section 6.6 that task based user models indeed result in better performance than basic bag-of-term based or basic topical interest based representation which further motivates us to investigate combining the two different modalities of user information: topical interests and tasks associations. Indeed, the information carried by user's topical interest profiles and their task profile are different and it would make sense to couple both these informations to jointly learn user profiles. In the next section, we further augment our task based user profiles by incorporating user's topical interest profiles and describe our tensor based approach for the joint model.

6.4 Combining Search Tasks with Topics

Our objective in this section is to build succinct user profiles from the search task information embedded in search logs while at the same time incorporating user's topical interest profiles. Building upon on prior work, we augment our task based

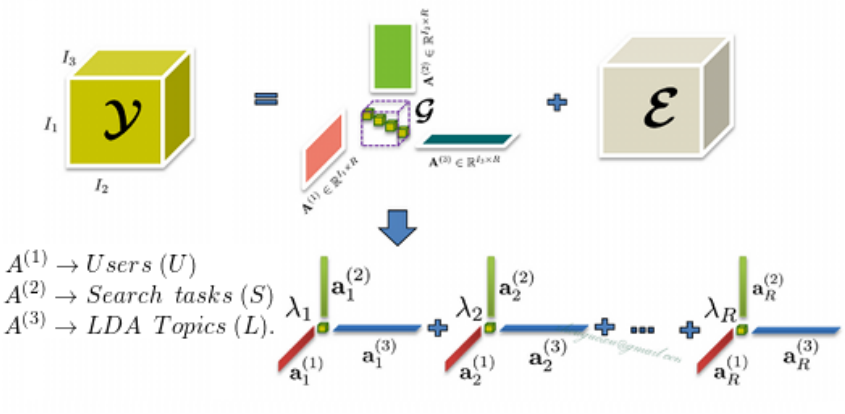


Figure 6.1: The overview of the user-topic-task tensor constructed by jointly considering user's topical interest profiles alongwith their search task interaction behavior. The tensor decompisiton breaks the tensor into latent factors which encode the complex interactions between the three different modes of the tensor.

user representation model with user's topical information by coupling the topical interest with task based information in the form of a tensor and learning user profiles based on the decomposition of the $< user, topic, task >$ tensor. We first describe the model we use for identifying topical interests of users and further show how we combine this model with task based representation.

6.4.1 Learning Topical Interest Profiles

Topical interests based methods are quite popular in learning user representations [93, 92]. Given user's history of search queries, we aim to develop a topic interest model which captures user's interest distribution over different topics. We make use of the Latent Dirichlet Allocation (LDA) model to learn the latent set of topics embedded in the search log [92]. It is to be noted that LDA topic model based approaches are standard methods to extract user's topical interest profiles and are widely used across user modelling applications.

We hypothesize that each search query is motivated by choosing a topic of interest first and subsequently a query is issued to describe that search need from the catalogue of words consistent with that particular topic. Based on this intuition, we learn an LDA based topic model and use the learnt model to do topical inference for each user to obtain a topic-distribution for the user over the set of learnt topics. We refer to this distribution as a user's ***topical profile***.

6.4.2 Coupling Topics & Tasks

Our main intuition behind leveraging both the topical profile as well as the search task profile of users is to better differentiate between users who share similar topical profiles. Topics and tasks capture different information: topical interest information help in capturing the user heterogeneity resulting from varied interests while task information helps in capturing user heterogeneity resulting from different information needs.

We formulate this intuition in our model by coupling task information with topical information on a per-user basis. We construct a 3-mode tensor $< user, topic, task >$ to jointly capture user's topical as well as search task based information. Next, we briefly describe the tensor formulation.

Tensors: a primer

A tensor is a multidimensional array. More formally, a N-way tensor or N-th order tensor is an element of the tensor product of N vector spaces each of which as its own co-ordinate system. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. The *order* of a tensor is the number of dimensions, also known as *modes*. A third order tensor can be represented as $T \in \Re^{I_1 \times I_2 \times I_3}$ with each element of the tensor denoted as $t_{i,j,k}$ with $i \in (1, I_1)$, $j \in (1, I_2)$ and $k \in (1, I_3)$. The symbol $\circ$ represents the vector outer product.

Constructing $< user, topic, task >$ Affinity Tensor

To jointly model the user's topical and task preferences, we construct a 3-mode tensor - users, topics and tasks. Each element of our tensor ($T \in \Re^{I_1 \times I_2 \times I_3}$), $t_{i,j,k}$ defines user i's combined task based and topical preference - a user's participation in a certain task gets weighted by his topical affinity, thereby coupling his task based and topical affinity. More formally, we define each tensor-component value as follows:

$$t_{i,j,k} = U_{i_{topic_j}} \times U_{i_{task_k}} \tag{6.4}$$

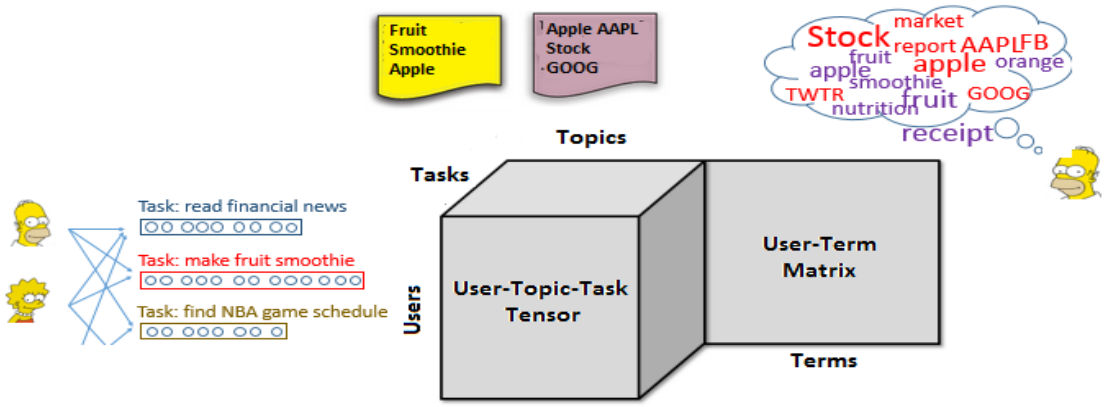


Figure 6.2: The coupled matrix-tensor obtained by coupling user's term usage behavior matrix with the user-topic-task tensor. The matrix and the tensor share a common mode of 'users'. On the left, we highlight some task related activity of the users and the associated topics obtained and the terms used on the top and right parts of the figure respectively.

where $U_{i_{topic_j}}$ is user U_i's topical affinity for topic j obtained from the LDA model learnt before while $U_{i_{task_k}}$ represents the task affinity for user U_i's for search task k obtained in earlier the user-task association phase (Section 6.3). To obtain user's topical affinity estimates (U_i), we train an LDA topic model on the entire query collection and use user's historical queries to create user's term profile which is then used for estimating the topic proportions using LDA inference techniques. I_1, I_2, I_3 are the different dimensions of the different modes of the tensor - in our case, these represent the number of users, number of topics and the number of search tasks extracted respectively. Thus, for each user we construct his coupled task-topic affinity value and populate the corresponding component in the tensor T.

Tensor Decomposition

Tensor decomposition methods are regarded as higher-order equivalents to matrix decompositions. The PARAFAC tensor decomposition [148] allows us to leverage connections between the different users across different topics and different search tasks. By PARAFAC, the input tensors are transformed into Kruskal tensors, a sum of rank-one-tensors. Formally, the tensor $T \in \Re^{I_1 \times I_2 \times I_3}$ is decomposed into component matrices $U \in \Re^{I_1 \times d}$, $T \in \Re^{I_2 \times d}$ and $S \in \Re^{I_3 \times d}$ and d principal factors

λ_i in descending order. Via these, tensor T can be written as a Kruskal tensor by:

$$T \approx \Sigma_{k=1}^{d} \lambda_k \cdot U^k \circ T^k \circ S^k \tag{6.5}$$

where λ_k denotes the k-th principal factor. The goal is to compute a decomposition with d-components that best approximates our tensor T, i.e., to find

$$min_{\widetilde{T}} \|T - \widetilde{T}\| \tag{6.6}$$

such that

$$\widetilde{T} = \Sigma_{k=1}^{d} \lambda_k \cdot U^k \circ T^k \circ S^k \tag{6.7}$$

We make use of the Alternating Least Squares (ALS) approach [149] to solve the above objective - having fixed all but one matrix, the problem reduces to a linear least-squares problem.

Overall, the above formulation helps us to couple user's topical interests with their search task associations and learn a user representation based on this coupled tensor. This tensor decomposition based user modelling approach allows us to use multi-modal user information and leverage insights from each of them while learning user representations.

Similar to other works based on tensors, an important characteristic of the proposed user modelling approach is that this method is generic enough and allows us to plug-in other sources of user information - click models, data from advertisement responses, etc.

6.5 Incorporating Historical Behavior

One widely used aspect of user behavior that provides especially strong signals for delivering better personalized services is an individual's history of queries and clicked documents. To construct the profiles necessary for personalization, evidence of a user's interests can be mined from observed past behaviors which can be sourced from their short-term (e.g., the current search session) or the long-term (e.g., across many previous sessions) search histories [93]. User's term history com-

prises of the set of terms users used to compose search queries. The tensor based approach described in the previous section looks at utilizing user's topical interest profile along with user's task association information. We hypothesize that additional signals about user's profile could be obtained by jointly modeling user's term usage behavior together with their task and topical interests information.

Overall, our motivation is to combine user's historic term usage behavior with their topical and task based information to learn user representations. We construct a user's term usage behavior over a set of combined vocabulary space. Combining the different users term histories together provides us with a user-term matrix (W), which we intend to jointly factorize while performing tensor factorization of the user-topic-task tensor (T). The idea behind the coupled matrix-tensor decomposition is that we seek to jointly analyze T and M, decomposing them to latent factors who are coupled in the shared user dimension. More specifically, the first mode of T shares the same low rank column subspace as M; this is expressed through the latent factor matrix U which jointly provides a basis for that subspace.

6.5.1 Coupled Matrix-Tensor Factorization (CMTF)

In the topic-task tensor we described earlier, we have a user by topic by task tensor which encodes user's topical interest profiles and task activities. We also have a semantic matrix which provides additional information for the same sets of users - the user by term matrix. In such cases, we may say that the tensor and the matrix are *coupled* in the *user* mode. Following Acar *et al.* [176], we next describe the joint analysis of a matrix (M) and a 3th-order tensor (T) with one mode in common, where the tensor is factorized using the CP model and the matrix is factorized by extracting latent factors using matrix factorization.

Let $T \in \Re^{I_1 \times I_2 \times I_3}$ and $M \in \Re^{I_1 \times I_4}$ have the first mode (user) in common; the objective function for coupled analysis is defined by [176]

$$f(U,S,L,W) = \frac{1}{2}\|T - [U,L,S]\|_F^2 + \frac{1}{2}\|M - UW^T\|_F^2 \tag{6.8}$$

Our goal is to find the matrices U, L, S, W that minimize this objective. In

order to solve this optimization problem, we can compute the gradient and then use any first-order optimization algorithm [178]. Rewriting the equation,

$$f(U,S,L,W) = f_1 + f_2 \tag{6.9}$$

where $f_1 = \|T - [U,L,S]\|_F^2$ and $f_2 = \|M - UW^T\|_F^2$. The partial derivative of f_1 with respect to the different matrices has been derived in [179] so we just present the results here. Let $Z = [U,L,S]$, then

$$\frac{\partial f_1}{\partial U} = (Z_i - T_i)U^{(-i)} \tag{6.10}$$

where $U^{(-i)} = U^{(I_1)} \odot \ldots U^{(i+1)} \odot U^{(i-1)} \odot \ldots \odot U^{(1)}$. Similar computations can be made for the other matrices components L and S. The partial derivatives of the second component, f_2, with respect to U,L,S and W can be computed as

$$\begin{aligned} \frac{\partial f_2}{\partial U} &= -MW + UW^TW \\ \frac{\partial f_2}{\partial U} &= -W^TU + WU^TU \end{aligned} \tag{6.11}$$

Combining the above results, the partial derivative of f with respect to factor matrix can be computed as

$$\begin{aligned} \frac{\partial f}{\partial U} &= \frac{\partial f_1}{\partial U} + \frac{\partial f_2}{\partial U} \\ \frac{\partial f}{\partial W} &= \frac{\partial f_2}{\partial W} \end{aligned} \tag{6.12}$$

Similar computations can be made for the S and L components. With these gradients, the aforementioned coupled matrix-tensor optimization problem can then be solved using any first-order optimization algorithm [176, 178].

On solving the coupled factorization objective[1], we obtain latent factor matrices which could be used as latent representations. More specifically, by making use of the latent factor matrix U we're able to learn user representations that jointly

[1]We make use of the CMTF toolbox provided by [176]: `http://www.models.life.ku.dk/joda/CMTF_Toolbox`

express user's topical, task and term profile information.

6.6 Experimental Evaluation

In order to evaluate the performance of the proposed user modelling techniques, we use three different techniques of evaluation based on collaborative query recommendation, query recommendation based on user groups and user cohort analysis.

6.6.1 Compared Approaches

We consider the following baselines to evaluate the performance of the proposed tensor based method:

- ***TermSim*** (TermSim) is a method that only uses bag-of-words based representation for each user where the terms are extracted from user queries and similar users found using cosine similarity between each user's bag-of-word based representations[90].

- ***LDA Topic Based*** (LDA) is a method of representing users in terms of their topical interests where the topics are extracted via a common Latent Dirichlet Allocation setup [92]. It s important to note that topic based representations are one of the most commonly used representations for personalization.

- ***Task Based:*** (Task) The first step towards coupling tasks with topics is representing users just in terms of search tasks. We use the user representations obtained in Section 3 as a result of matrix factorization as another baseline to compare the gain in performance obtained as a result of adding the topical aspect on top of user's search task information [175].

- ***TT-Tensor:*** (TT) Topic-Task Tensor (TT-Tensor) based user representation is the proposed technique which combines user's task information with their topical interests.

- ***CMTF:*** Coupled Matrix Tensor Factorization (CMTF) [176] based user representation is our second novel contribution which takes into account the user histories in addition to their topical and task based profiles.

User Profile Information	TermSim	LDA	Task	TT-Tensor	CMTF
Term History	✓				✓
Topical Interests		✓		✓	✓
Search Task information			✓	✓	✓

Table 6.2: User profile information encapsulated in each of the compared approaches. We notice that the proposed TT-tensor and CMFT based methods maximally incorporate the different user profile information available.

Each of the compared approaches work with different user information. In Table 6.2 we summarize the different modalities of user information used by the different approaches.

6.6.2 Dataset

We make use of the AOL log dataset which consists of ∼20M web queries collected over three months and use data for a subset of ∼1200 users who have issued more than certain threshold (550) number of queries. We run our Task Discovery algorithm on the set of queries for each of these users which results in a total of ∼0.12M tasks which we cluster to obtain a set of 1521 search tasks. Such a setting for task extraction is in line with the original proposed research by Lucchese *et al.* [70]. These tasks are then used to create the user-task association matrix, as described in Section 6.3 and for constructing the coupled matrix-tensor, as described in Section 6.5. To make fair comparisons between the topical and task based user profiles, we keep the number of latent factors for tasks same as the number of latent topics.

6.6.3 Collaborative Query Recommendation

A good user profile for query recommendation should capture a user's specific interests and informational needs. Based on this intuition, we evaluate performance of the proposed approach on ***Collaborative Query Recommendation*** [95] where the goal is to recommend queries to a user based on queries issued by similar users. For each user we select the n-most similar users where the similarity is calculated by a cosine similarity score using the user representations learnt. We calculate the weighted frequency of a candidate query for most similar users of the target user u, and select the top-k queries as recommendation.

To evaluate the performance of the above mentioned techniques, we consider

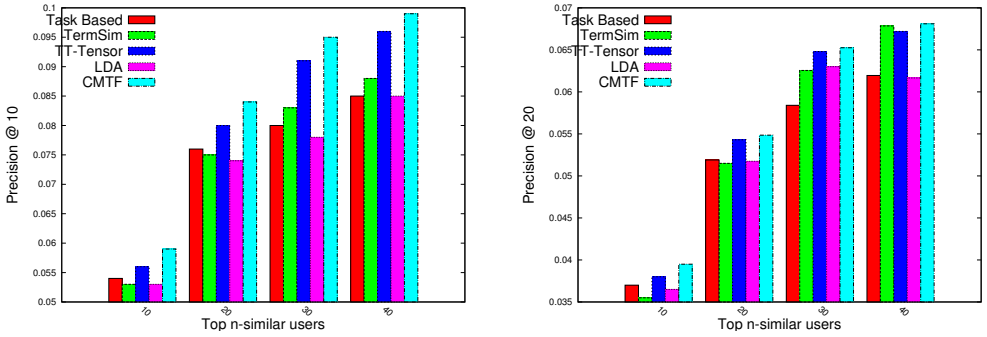


Figure 6.3: Performance on Collaborative Query Recommendation (left figure: Precision@10 & right figure: Precision@20). Based on the average number of query matches between the recommended set of queries and user's own test set of (unseen) queries, the precision at 10 and precision at 20 values are plotted against the number of similar users considered (n). The results obtained at n=10, 20, 30 (left) and n=10, 20 (right) were statistically significant ($p<0.05$) based on pairwise tests between the proposed method and the best performing baseline.

the test-set of queries in the target user as relevant, and computed average number of relevant queries matched in the recommendation query set as the performance metric. The training/test set per user is populated based on a 20% split across all user queries. We use the training set for populating the matrix/tensor while the test set of queries per user for evaluating the quality of the recommended queries. We plot precision@10 and precision@20 values based on the average number of query matches between the recommended set of queries (top-10 (left) and top-20 (right)) and user's own test set of (unseen) queries. Given that the task at hand is collaborative query recommendations from similar users, comparison with other general purpose query recommendation techniques is beyond the scope of our experiments.

Discussion

Our results (Figure 6.3) show that the proposed Topic-Task Tensor based user modelling approach(*TT-Tensor*) and the coupled matrix factorization method (CMTF) performs better than *TermSim* as well as *TaskBased* which demonstrates that combining search task information with user's topical interests thus help us better capture different aspects of user profiles and can serve as potent user modelling tools. Since *TermSim* relies strictly on term matching for measuring user similarities, its coverage is limited: it might not capture insights for the users with too few queries

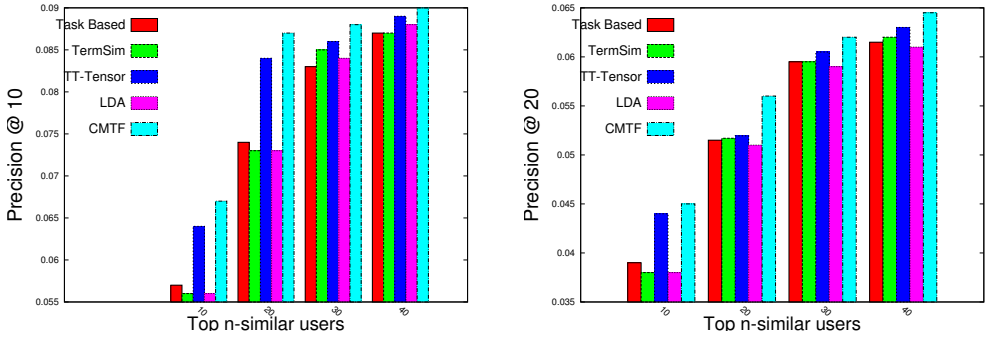


Figure 6.4: Performance on Cohort Query Recommendation (left figure: Precision@10 & right figure: Precision@20). Based on the average number of query matches between the recommended set of queries and user's own test set of (unseen) queries, the precision at 10 and precision at 20 values are plotted against the number of similar users from user's cluster considered (n). The results obtained between the CMTF and the best performing baseline at n=10, 20 (left) and n=10, 20, 30 (right) were statistically significant ($p<0.05$) based on pairwise tests between the proposed method and the best performing baseline.

or those who shared the same search interest but issued different queries or performed different tasks. Task based user modelling can help in better differentiating between users which have similar topical interests but perform different tasks.

The proposed tensor based approach combines the best of both the worlds and hence was able to leverage the topical user profile information with the task aspect. Additionally, the CMTF model combines information from all available data modalities and learns a joint user representation. We see that the CMTF model outperforms the other methods which highlights the importance of jointly considering user's term, topic and task information. On analysis of the dataset, we figured out that the overall lower average query recall values across all methods can be attributed to the fact that there is less query overlap between users, i.e., the upper limit of common query among users is indeed low on average.

6.6.4 Cohort based Query Recommendation

It is well-known that preferences across a user population often decompose into a smaller number of communities of commonly shared preferences [180, 181]. In this study, we investigate the performance by means of *groupization*: a variant of personalization whereby other users' profiles can be used to personalize current user's experience. As opposed to finding similar users from the entire user popu-

lation for collaborative query recommendation, we explore the use of user-cohorts obtained above and leverage information from users belonging to the same cluster to aid in query recommendation. A good cluster should contain better similar users - users who are indeed more representative of the current user. Based on this, we evaluate the performance of the proposed approach on Cohort based Query Recommendation where the goal is to recommend queries to a user based on queries issued by users in the same cluster. Following similar set up as before, we present cohort-based query recommendation results (clustering performed with 10 clusters) in Fig. 6.4.

Discussion

The proposed approach of encapsulating user's historic term usage behavior with their topical and task oriented interests consistently performs better than our baselines in terms of recommending queries from users from the cluster. As can be seen in Fig. 6.4, the CMTF and coupled task-topic representation performs significantly better right at the start with the difference between the approaches slimming down as we go towards more query recommendations. This is indeed expected since we are measuring precision of queries and eventually not-so-efficient methods will eventually be able to recommend better queries as we increase the number of queries suggested.

Recent research on groupization has focussed on developing different ways of building user cohorts based on topical interests, location, etc [168]. In the present study, we used simple clustering on user features for building cohorts; in future study we intend to compare cohorts of varying sizes and variants of cohort construction techniques to obtain detailed insights on user cohort behaviors.

In addition to performing cohort based query recommendation, we also investigate the *goodness* of the user cohorts we obtain, which were used for query recommendation as described above. We next describe the experimental set-up to analyze the performance of the compared approaches on the task of user cohort formation.

	DB Index					SI Index				
nClusters	**TermSim**	**LDA**	**Task**	**TT**	**CMTF**	**TermSim**	**LDA**	**Task**	**TT**	**CMTF**
10	1.61	1.55	1.98	1.52	**1.46**	0.19	0.20	0.16	0.43	**0.48**
30	1.69	1.66	1.83	**1.48**	1.47	0.23	0.26	0.24	0.36	**0.41**
50	1.58	1.65	1.84	1.52	**1.50**	0.27	**0.28**	**0.28**	0.27	0.27
80	1.71	1.67	1.80	1.58	**1.57**	0.29	0.35	0.28	0.47	**0.51**
100	1.75	1.65	1.76	1.63	**1.59**	0.31	0.57	0.32	0.58	**0.62**

Table 6.3: Cluster Analysis of User Representations - cluster evaluation metrics performance for the different approaches are shown. *TermSim* represents the simple term similarity baseline, *LDA* represents the topic model based user representations, *Task* represents user representations learnt via PMF by using task information while *TT* represents the proposed Task-Topic Tensor based user representations.

	CH Index				
nClusters	**TermSim**	**LDA**	**Task**	**TT**	**CMTF**
10	453	643	352	534	**658**
30	297	353	203	377	**411**
50	213	258	151	285	**299**
80	178	192	116	212	**234**
100	96	165	99	182	**194**

Table 6.4: Cluster Analysis of User Representations - internal cluster evaluation metric (CH Index) performance for the different approaches are shown. *TermSim* represents the simple term similarity baseline, *LDA* represents the topic model based user representations, *Task* represents user representations learnt vi PMF by using task information while *TT* represents the proposed Task-Topic Tensor based user representations.

6.6.4.1 User Cohort Analysis

We believe that incorporating task behavior of users while learning user representations enables us to better *decompose* users into user cohorts or clusters. In this study, we test the hypothesis that a good user modeling scheme would allow for good cluster formation based on the learnt user representations. We evaluate the user representations learnt in terms of the quality of user clusters formed. Unlike external cluster validation measures, which use external information ("true" cluster membership) not present in our data, internal cluster validation measures only rely on information in the data [182]. In Table 6.3 and Table 6.4, we present the cluster validation results on a variety of different metrics, which, to the best of our knowledge, represent a good coverage of the validation measures available in different

fields, such as data mining, information retrieval, and machine learning.

The different measures used capture different *goodness* measures of clusters based on inter-cluster and intra-cluster similarities. The Davies-Bouldin index (DB) [183] is calculated as follows. For each cluster C, the similarities between C and all other clusters are computed, and the highest value is assigned to C as its cluster similarity. Then the DB index can be obtained by averaging all the cluster similarities. The smaller the index is, the better the clustering result is.

The Silhouette index (SI) [184] validates the clustering performance based on the pairwise difference of between and within-cluster distances. The Calinski-Harabasz index (CH) [185] evaluates the cluster validity based on the average between and within cluster sum of squares. Larger values of SI and smaller values of CH are preferred.

Discussion

As can be seen in Table 6.3 and table 6.4, the user clusters obtained from via using topic-task coupled representations indeed perform better than the clusters obtained via just Bag-of-Terms or task baselines. This is in line with our hypothesis that capturing task behaviors across user populations indeed helps us in forming *well-knit* user clusters and thus could help us perform better in "*groupization*". Having good clusters could be useful for many applications, one of them being collaborative query recommendation, as shown above.

6.7 Conclusion

We presented a novel approach to couple user's topical interest information with their search task information and their term usage behavior to learn a joint user representation technique. We demonstrated that coupling user's task information with their topical interests indeed helps us build better user models. We show through extensive experimentation that our task based method outperforms existing query term based and topical interest based user representation methods. This clearly demonstrates the value of considering *search tasks* rather than just query terms or topics

during personalization. Future work involves the development of more sophisticated and generalizable models of task behavior that can model task-relevant activity beyond search engine interactions. The flexibility of the tensor based framework makes our proposed method generic enough to add more data sources and modalities. The user representations learnt can be used for various different applications, something we intend to explore as future work.

Chapter 7

Learning Query Embeddings using Task Context

7.1 Introduction

Users tend to seek information by issuing queries to a search engine. The need for search often resides within an external context that prompts the user to formulate their information needs as search queries. When an information need, or task, requires multiple searches, the sequence of queries form a context which influences interaction behavior for the duration of the search process. Search context plays an important role in understanding user's needs and can be leveraged to develop better representations and ranking models. While a major portion of existing work have investigated user behavior using search sessions as the fundamental focus of search activity, search tasks are emerging as a competing perspective in this space with recent studies suggesting that users seek to complete multiple search tasks within a single search session, while also taking multiple sessions to finish a single task at times [16]. As a result, search tasks have steadily emerged as accurate units to capture searcher's goals and seeking behavioral insights.

A direct result of users being engaged in multitasking and task switching behaviors is that the resulting search context is heterogeneous, composed of interleaved search goals and tasks. Recent advancements in task extraction techniques have made it possible to segregate search activity logs into a set of interleaved tasks

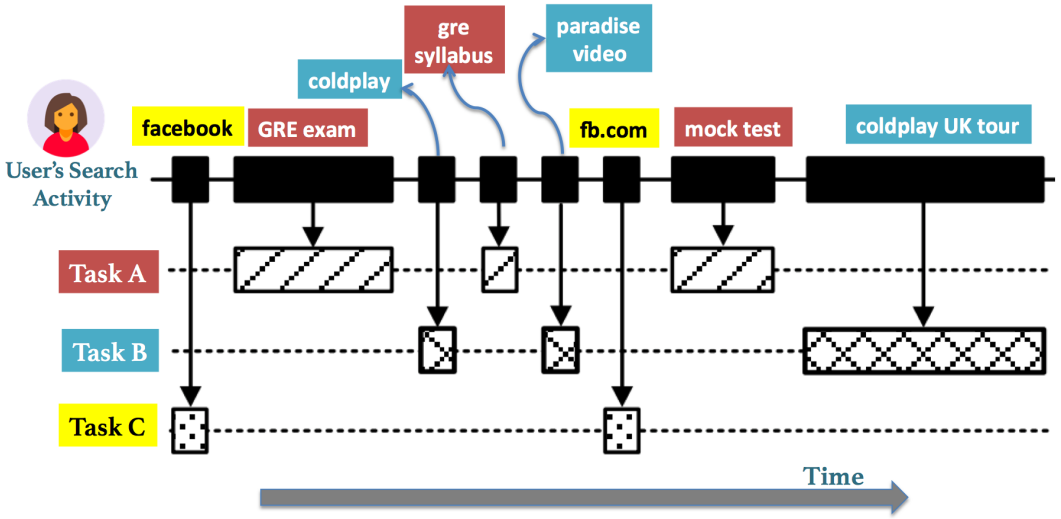


Figure 7.1: Exemplar user interaction with search engine.

[165, 186]. For example, while multi-tasking in their search sessions, users pursue many different tasks at once, often switching between them. Figure 7.1 shows an example of a user multitasking in her search session, with her task of finding information on GRE exams interleaved with finding music videos of Coldplay band. Such heterogenous context makes it difficult for the retrieval system to use to create localized user interest models, provide contextual result rankings, query suggestions, and other user support offerings.

In this chapter, we aim at mitigating the ill effects of heterogeneous contexts by leveraging task information while learning representations of users' information needs. Learning meaningful and accurate representations of queries is an important problem in web search, with most retrieval, ranking, query expansion and query suggestion methods heavily relying on informative ways of representing search queries. Beyond traditional one-hot vectors and TF-IDF approaches, the distributed semantic representations based on dense vectors of vocabulary terms, also known as word embeddings, have been shown to be highly effective in many natural language processing and information retrieval tasks [187, 129]. In general, these approaches provide global representations of words; each word has a fixed representation, regardless of any discourse context. While a global representation provides some advantages, search context can vary dramatically by task. Most word embedding

learning techniques rely on a window-based training which uses local information in immediately surrounding words. Given the multi-tasking nature of search sessions, the resulting query context is rendered heterogeneous and might consist of queries from different unrelated tasks.

In this work, we aim at leveraging task context while learning query representations. Specifically, we propose a novel task based embedding architecture to learn distributed semantic representation of query terms which prefers task context over local information in immediately surrounding words. We propose that embeddings be learned on a task-constrained context instead of the traditionally used global or session context. The proposed task embedding model is able to extract improved query representations which capture task context. In addition to qualitative analysis, we demonstrate the benefit of learning task based embeddings over traditional query representation techniques by showing enhanced performance when generating query suggestions. Our findings have implications on the design of future task aware search systems which better model user needs and help them in accomplishing their task.

7.2 Task Embeddings

Our goal in this work is to learn richer embeddings and explore the use of task context to learn more contextual query representations. In this section, we propose a novel embedding architecture based on task context. As a precursor to generating relevant task context, we first need to extract the set queries which belong to the same overall task given a sequence of queries issued by a user over a period of time.

7.2.1 Extracting "*On-Task*" Queries

In order to extract *on-task queries*, we make use of the Latent Structural SVM framework [10] for task identification. Given query sequences, search tasks are identified by clustering queries into tasks by find the strongest link between a candidate query and queries in the target cluster (*bestlink*). We run the task extraction algorithm as described in section 2.4.4 of Chapter 2 on search logs to extract all queries belonging to the same task. Such a query collection is henceforth referred

to as "*on-task queries*". We run the task extraction algorithm on search logs to extract all queries belonging to the same task. Such a query collection is henceforth referred to as "*on-task queries*".

7.2.2 Task Context Embedding Architecture

Estimating accurate query representations plays a crucial role in many information retrieval tasks and past work have relied on a number of different ways of building such representations from simple term frequency based approaches to the recent word embeddings. While generically learnt word embedding models have performed well in various NLP tasks, we hypothesize that incorporating task context while learning query embeddings would result in more accurate representation. In this section we describe the propose task based embedding architecture which leverages the task information as described in Section 7.2.1.

Given a search log comprising of a set $\boldsymbol{S}$ of $\|S\|$ query sequences obtained from online users, where each query sequence $S = (q_1, ..., q_{M_s}) \in \mathbf{S}$ is defined as an uninterupted sequence of M_s queries, and each query $q_m = (w_{m1}, w_{m2}, ...w_{mT_m})$ consists of T_m words, our objective is to find D-dimensional real-valued representation $v_{q_m} \in R^D$ of each query q_m. We begin by tagging task membership information for each query t_{q_m} using the task extraction module and casting a query sequence from a given user as a *sentence* fed into the neural embedding model.

Traditionally, embedding based models learn query representations using the skip-gram model [122] by maximizing the objective function over the entire set S of search sessions, defined as:

$$L = \sum_{s \in S} \sum_{q_m \in s} \sum_{-b \leq i \leq b, i \neq 0} logP(q_{m+i}|q_m) \tag{7.1}$$

where v_q and v_q^i are the input and output vector representations of query q, b is defined as length of the context for query sequences, and V is the number of unique queries in the vocabulary. To incorporate the task context, we modify the objective function and incorporate a selective task context window selection function in the

likelihood objective:

$$L = \sum_{s \in S} \sum_{q_m \in s} \sum_{-b \leq i \leq b, i \neq 0} \mathbb{1}(t_{q_{m+i}} = t_{q_m}) \times log\ P(q_{m+i}|q_m) \tag{7.2}$$

The objective only considers surrounding queries which belong to the same task as the current query and disregards other non-task queries from consideration for a query's context. An alternate approach could replace the exact task based matching of queries ($\mathbb{1}(t_{q_{m+i}} = t_{q_m})$) by a probabilistic matching, which we leave for future investigation. Probability $P(q_{m+i}|q_m)$ of observing a neighboring query q_{m+i} given the current query q_m is defined using soft-max,

$$P(q_{m+i}|q_m) = \frac{exp(v_{q_m}^T v'_{q_{m+i}})}{\sum_{q=1}^{|V|} exp(v_{q_m}^T v'_{q_m})} \tag{7.3}$$

From these equations we can see that the model considers the temporal and task context of query sequences, where queries with similar contexts (i.e., with similar neighboring queries which belong to the same task) will have similar vector representations in the projected semantic space.

The proposed objective is optimized using stochastic gradient ascent, suitable for large-scale problems. However, computation of gradients $\bigtriangledown L$ for the likelihood function equation above is proportional to the vocabulary size V , which is computationally expensive in practical tasks as V could easily reach hundreds of millions. As an alternative, we used negative sampling approach proposed in [122], which significantly reduces the computational complexity.

7.3 Experimental Evaluation

In this section we demonstrate the benefit of incorporating task context while learning query representations. We consider the task of query suggestions and provide empirical comparisons of the proposed method against various baselines.

7.3.1 Dataset

In order to extract query embeddings, we use a random sample of 1 week of search log data from May 2016 of a commercial web search engine comprising of user ID information along with session identifier and query text. The dataset composed of over 24M search impressions spread over 8M search sessions, issued by over 200K users, resulting in a vocabulary size of over 5M words. We train the Continuous-Bag-of-Words model of Word2Vec using all the queries in this corpus. As per the free parameters, the dimension of the word vectors was set to values in 100, 300, the number of negative examples is in 5. Since query text is used to learn embedding, we keep the window size as 2 which totals to 4 words as context per query term. Sub-sampling of frequent terms was not performed and all other parameters were set to default values.

7.3.2 Baselines

We consider a number of baselines, including non-neural approaches as well as neural embedding based approaches.

1. One-hot vector representation: The traditional representation technique which represents queries using 0/1 vector encoding.

2. Global Embeddings: We use word2vec model trained on GoogleNews corpus as global embeddings.

3. Session embeddings: We use search sessions as context while learning embeddings.

4. Random: To validate the usefulnes of considering query sequence information, we randomly shuffle queries issued by a user before inputting to embedding learning model.

7.3.3 Qualitative Analysis

We begin with a qualitative analysis of the extracted representations by showing *nearby* query terms. Table 7.1 shows the top 3 query terms which are most similar

Query: london		Query: usps	
Global	**Task Context**	**Global**	**Task Context**
birmingham	weather	postal_service	track
nyc	time	fedex	hours
england	tube	track	delivery

Table 7.1: Qualitative comparison of similar words fetched using global embeddings and task embeddings.

	TREC Tasks 2015		
Method	**NDCG@3**	**NDCG@5**	**NDCG@10**
Random	0.613	0.56	0.542
Global	0.631	0.572	0.558
Session	0.633	0.573	0.564
Task	**0.662**$^{*\&}$	**0.591**$^{*\&}$	**0.571**$^{*\&}$

Table 7.2: Average Relevance results. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the Global and Session context baselines respectively.

	TREC Tasks 2016		
Method	**NDCG@3**	**NDCG@5**	**NDCG@10**
Random	1.13	0.99	0.926
Global	1.15	1.01	0.932
Session	1.14	1.02	0.934
Task	**1.16**$^{\&}$	**1.04***	**0.944**$^{*\&}$

Table 7.3: Average Relevance results. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the Global and Session context baselines respectively.

to two randomly chosen queries. We observe that the suggestions shown using task embeddings are more coherent and related than the ones from global embeddings. In web search context, suggestions like '*weather*' and '*tube*' are more contextually relevant to be one of the aspects a user might be looking for rather than suggestions comprising of similar city name suggestions.

In addition to the qualitative analysis, we provide empirical evidence which demonstrates the usefulness of considering task context when learning representations.

	TREC Tasks 2015		
Method	**NDCG@3**	**NDCG@5**	**NDCG@10**
Random	0.289	0.3	0.302
Global	0.292	0.301	0.309
Session	**0.314**	0.309	0.318
Task	**0.314***	**0.311**$^{*\&}$	**0.321**$^{*\&}$

Table 7.4: NDCG@k results. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the Global and Session context baselines respectively.

	TREC Tasks 2016		
Method	**NDCG@3**	**NDCG@5**	**NDCG@10**
Random	0.511	0.509	0.522
Global	0.524	0.513	0.52
Session	0.510	0.514	0.527
Task	**0.526**$^{\&}$	**0.529**$^{*\&}$	**0.535**$^{*\&}$

Table 7.5: NDCG@k results. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the Global and Session context baselines respectively.

7.3.4 Query Suggestions

To evaluate our approach we make use of the query representations obtained to generate lists of query suggestions. Specifically, we make use of the TREC Tasks track data from two years (2015 & 2016) to rank query suggestions that would help users fulfil their information need. Tasks track data provides test collections for evaluating the usefulness of retrieval systems in terms helping people achieve their search tasks. The dataset comprise of 100 different tasks embodied by a query each, with each task containing a list of possible candidate queries that represent the set of all tasks a user who submitted the query may be looking for. Each of these candidate queries are judged for relevance labels by human assessors. Overall, the dataset spans over 2 years and comprises of X candidates in 2015 and Y candidate queries in 2016.

For each query, we consider the entire pool of candidate queries and use the query representations to find the similarity of the query with each candidate query based on which they are rank. The relevance labels provided with each candidate query are used to compute the average relevance and NDCG@k metrics.

Tables 2 presents the average precision scores for the different baselines considered. We observe that though global representations perform better than traditional and simpler representation techniques, they perform worse than session based and task based embeddings. This highlights the importance of considering local context when learning representations, since generic contexts are usually heterogeneous and ill fitted to retrieval problems. Among the neural local context models, task based context performs better than session based contexts. This confirms our hypothesis that sessions are usually polluted with queries from various tasks, and as a result the resulting context isn't informative enough. An overall performance improvement of all compared approaches is observed in the TREC Tasks 2016 dataset over 2015 dataset; this is in line with the performance improvement observed in the TREC submissions as well.

While relevance scores are important, often system designers have a constraint to rank top-k suggestions. To this end, in addition to average relevance scores, we make use of the candidate ranking to compute NDCG scores and present results in Table 3. Similar to our previous observation, we observe that neural representation methods generally perform better than non-neural models. Amidst session based and task context based, task based representation performs better than the corresponding session context.

7.4 Conclusion

Search context has played an important role in solving various retrieval tasks. In this chapter, we leveraged task context to learn query representations. Experimental evidence suggests tasks context enriched representations perform better than traditional representations, and at the same time, task context is more informative than session context. These findings have implications in designing better personalization and recommendations techniques aimed at exploiting task context for enhanced support.

Chapter 8

Deep Sequential Models for Task Satisfaction Prediction

8.1 Introduction

As search systems have advanced, an increasingly larger proportions of users are relying on search engine to satisfy their information needs. Developing better understanding of how users interact with search engines is becoming important for gauging user satisfaction and improving user's search experience. Since obtaining explicit feedback from users is often prohibitively expensive and challenging to implement in real-world systems, commercial search engines have exploited implicit feedback signals derived from user activity. While users interact with a search engine, they leave behind fine grained traces of interaction signals. These interaction signals contain valuable information, which could be useful for predicting user satisfaction as well as developing metrics for search engine evaluation to assist rapid experimentation.

User initiated search is often motivated by a search goal, or a task. A simple task refers to an atomic information need resulting in one or more queries [13]. Understanding and evaluating a search engine's performance from a task centric view attains paramount importance. Most existing work on gauging user satisfaction have focused on query level satisfaction [188, 100, 99, 119], with some initial efforts aimed at measuring task satisfaction for simple tasks [109]. Often, indepen-

dent information needs arise from an overall complex search task, where a *complex search task* refers to a multi-aspect or a multi-step information need consisting of a set of related tasks, each of which might recursively be complex [13, 9, 186, 19]. While existing work has primarily focused on measuring user satisfaction on simple search tasks, work on understanding and measuring user satisfaction for complex search tasks remains in its infancy.

In this chapter, we take a comprehensive look at user satisfaction from different levels of abstractions. We begin by investigating query level satisfaction, and propose a deep sequential model which considers holistic view of user's interaction with the search engine result page (SERP), constructs detailed interaction sequences of their activities and leverages such interaction sequences to predict query level satisfaction. In addition to interaction sequences, we consider various different behavior signals (e.g. click features, dwell times) and treat such signals as auxiliary features providing an alternate view of user interactions. We propose a unified multi-view deep model composed of parallel convolutional and recurrent neural networks capable of utilizing both the views of user interactions for predicting query level satisfaction. Finally, we go beyond query level abstraction and consider the problem of task satisfaction prediction. We propose a novel functional composition model which takes into account user satisfaction at the query level and the subtask level when making task satisfaction predictions. We present rigorous evaluation of the proposed approach using crowdsourced judgments as well as large scale pseudo-labeled data and demonstrate that the unified multi-view deep sequential model significantly outperforms a number of established baselines at query satisfaction prediction. We additionally show that the proposed deep sequential models are also better at predicting task level satisfaction. Our findings provide a valuable tool for gauging task satisfaction and developing next generation task-aware search engines.

Our work is different from existing work not only in measuring query level satisfaction but also in measuring task satisfaction. We not only consider extracting interpretable interaction sequences from user interactions, we propose novel ways

of combining different views of user interactions in a unified model for predicting query satisfaction. Further, unlike past work which considers simple tasks composed of query reformulations for measuring task satisfaction, we go beyond such simpler tasks and also consider complex tasks composed of many different queries and subtasks. Finally, we propose novel ways of aggregating query satisfaction estimates for task satisfaction prediction.

8.2 Problem Formulation

Our goal in this chapter is to extract and leverage user interaction data to predict query and task level satisfaction. We begin by defining the key concepts used throughout the chapter.

Sequence: Given a search impression and a list of possible user actions, a sequence is defined as a time-ordered list of actions performed by the user when interacting with the search result page.

Search Task: A search task is an atomic information need resulting in one or more queries [13]

Complex Task: A complex search task is a multi-aspect or a multi-step information need consisting of a set of related tasks, each of which might recursively be complex [113].

With this background, we formally define the problem of satisfaction prediction as:

Query Satisfaction (QSAT): Given user interaction data, predict whether the user's interaction with the search engine result page (SERP) rendered for the query was satisfying or not.

Task Satisfaction: Given a sequence of queries issued by the user to accomplish a complex task along with user interaction data and satisfaction estimate for each query, predict user's satisfaction in accomplishing the overall task.

In order to make task satisfaction predictions, we employ query level satisfaction estimates as well as subtask level satisfaction estimates. While few efficient approaches exist for identifying subtasks [186], we assume access to subtask demarcation information (obtained via crowdsourced labeling) for the scope of this

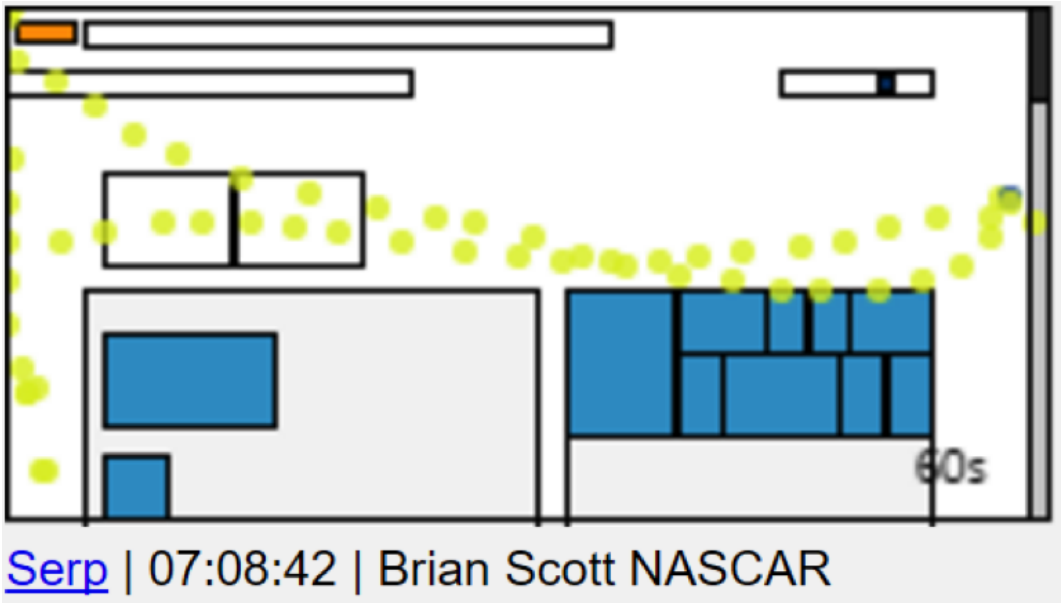


Figure 8.1: Example of user interaction with the SERP elements rendered for the query *Brian Scott NASCAR*. The sequence of green dots denotes the user's cursor position over a period of time.

chapter. We first describe the technique used to extract meaningful action sequences from user interactions with SERP (Section 8.3) . We then present in Section 8.4 our proposed deep sequential model for query level satisfaction prediction. Finally, in Section 8.5 we present different techniques for functional composition of query satisfaction estimates to make task satisfaction predictions.

8.3 Extracting User Interaction Data

The richness of the result page rendered in response to a user query allows users to interact with SERPs in myriad ways, including clicking results, scrolling, expanding task panels, hovering over images, pausing to read and absorb content among others. While most existing work has considered click based interaction signals or mouse movement features, these signals either lack coverage or are often abstracted at high SERP-level aggregates, which blinds the models to finer level user interaction signals. Our aim here is to analyze user interaction with the SERP (as depicted in Figure 8.1) and extract an interpretable interaction sequence. To do so, we construct a universal action sequence timeline from the following three different timelines:

1. **Viewport Timeline**: Viewport is defined as the position of the webpage that

is visible at any given time to the user. Viewport timeline allows us to consider user actions concerning the viewport, for example, scroll on the result page and resize of the screen.

2. **Cursor Timeline**: The cursor timeline provides us with all the cursor related user activity. Backend search logs record detailed user mouse activity which helps us to track the mouse movement and link the corresponding cursor activity to the different elements on the SERP.

3. **Keyboard Timeline**: The keyboard timeline records all keyboard related user activity (for example, text enter).

For each search impression, we log the three timelines with corresponding user actions along with the timestamp. Based on these three timelines, we generate one holistic universal action sequence timeline describing all user activity on the SERP by temporal sorting of individual timelines followed by stacking up the three timelines, and then interleaving them based on timestamps of the recorded actions. This provides us with a universal sequence of user interaction, examples of which are shown in Table 8.2. We next take a more detailed look at the actions considered to construct the timelines.

Actions Considered: In order to construct the three timelines, we considered a number of actions which include all types of interactions performed by the users. For click based actions, we associate the cursor information with the corresponding element on the SERP and recorded the joint action-element pair as an action, for example, click_algo1 signified a click on algorithmic result at position 1. Beyond clicks, we considered a range of cursor movement actions ranging from simple Move (denoting a mouse movement across different SERP element) to more sophisticated and intentional cursor movements like a *MouseRead*. We define a *MouseRead* as a horizontal line across a result snippet of length $>$ 50px and duration $>$ 100 ms that goes from left to right which starts and ends inside an algo-result, or advertisement or an answer result. Beyond cursor movement actions, we consid-

Action	Description
Click_algoX	Click on the X-th algorithmic result
Click_Ans	Click on any answer (non-image) result
Click_IMG	Click on any image result
MouseRead	horizontal line across a result snippet of length $>$ 50px and duration $>$ 100 ms that goes from left to right which starts and ends inside an algo-result, or advertisement or an answer result
Scroll	page scroll recorded on the search engine result page
Move	any cursor movement of length $>$ 10px and duration greater than $>$ 50 ms
pause	smallPause: no cursor movement on the SERP for time $<$ 5 seconds mediumPause: no cursor movement on the SERP for 5s $<$ time $<$ 20s longPause: no cursor movement on the SERP for 20s $<$ time $<$ 40s veryLongPause: no cursor movement on the SERP for time $>$ 40s
Resize	change in the size of the window/screen encompassing the result page
IssueQuery	user movement to the Search Box on the SERP and typing of text in the query box
dwellTime	smallDwellTime: dwell time on a clicked result URL with time spent $<$ 10s mediumDwellTime: dwell time on a clicked result URL with 10s $<$ time $<$ 40s longDwellTime: dwell time on a clicked result URL with time spent $>$ 40s
QuickBack	click on a SERP URL followed by returning back to the SERP within 5s

Table 8.1: Examples of actions considered along with their description used to create the user interaction sequence.

Example Sequences
Scroll $\rightarrow$ smallPause $\rightarrow$ Move-algo-1 $\rightarrow$ smallPause $\rightarrow$ Move-algo-2 $\rightarrow$ smallPause $\rightarrow$ Click-algo-2
smallPause $\rightarrow$ Move $\rightarrow$ Click-IMG $\rightarrow$ longDwell-Time

Table 8.2: Example of sequences extracted.

ered inter-activity time as pauses and categorized a pause into one of three types based on the duration of the pause: (i) short pause (time), (ii) medium pause (time) and (iii) longPause (time). We additionally considered issuing query and scroll related activities. Table 8.1 lists the major actions considered.

8.4 Query Level SAT Prediction

While implicit feedback measures like mouse clicks, reading and dwell times, gaze tracking have been extensively used in predicting search satisfaction, they ignore the sequence information accompanying any user interaction. Given the detailed action sequence extracted from user's interaction with SERP, we aim at predicting user satisfaction using the extracted sequence.

Gauging user satisfaction is the problem of predicting satisfaction label given a

query q, the search results page rendered, detailed user interaction actions recorded (Section 8.3) and aggregate implicit signals according to a parametric probability measure:

$$y = arg\ max_{y \in 0,1} p(y|q, SERP, \mathbf{A}, f; \theta) \tag{8.1}$$

where θ represent a vector of all parameters to learn, q is the query, $\mathbf{A}$ is the user action sequence. In order to predict query level satisfaction, we leverage interaction sequences and propose a deep sequential model to predict satisfaction. Further, we augment the sequence model with SERP level signals which have been traditionally used to propose a coupled model which combines interaction sequence information with auxiliary implicit feedback signals to propose a unified model for query level satisfaction prediction.

8.4.1 Sequential Model for SAT

To leverage the entire interaction sequence we make use of recent advancements in the field of deep recurrent network and formulate our problem as that of sequence classification. Recurrent neural networks (RNNs) are a powerful family of connectionist models that capture time dynamics via cycles in the graph, thereby enabling them to process sequences of data. A RNN maintains a memory based on history information, which enables the model to predict the current output conditioned on long distance features. An important characteristic of user interactions is that the resulting sequences are of variable length. Long Short-Term Memory (LSTM) networks are a special case of Recurrent Neural Networks (RNNs) which are capable of creating internal cell states of the network which allow it to exhibit dynamic temporal behavior thereby enabling the RNN to process arbitrary sequences of inputs such as user interaction sequences.

The action-LSTM takes as input a sequence of user actions $x = (x_1, x_2, ..., x_T)$ and computes the hidden sequence $h = (h_1, h_2, ..., h_T)$ as well as the output vector

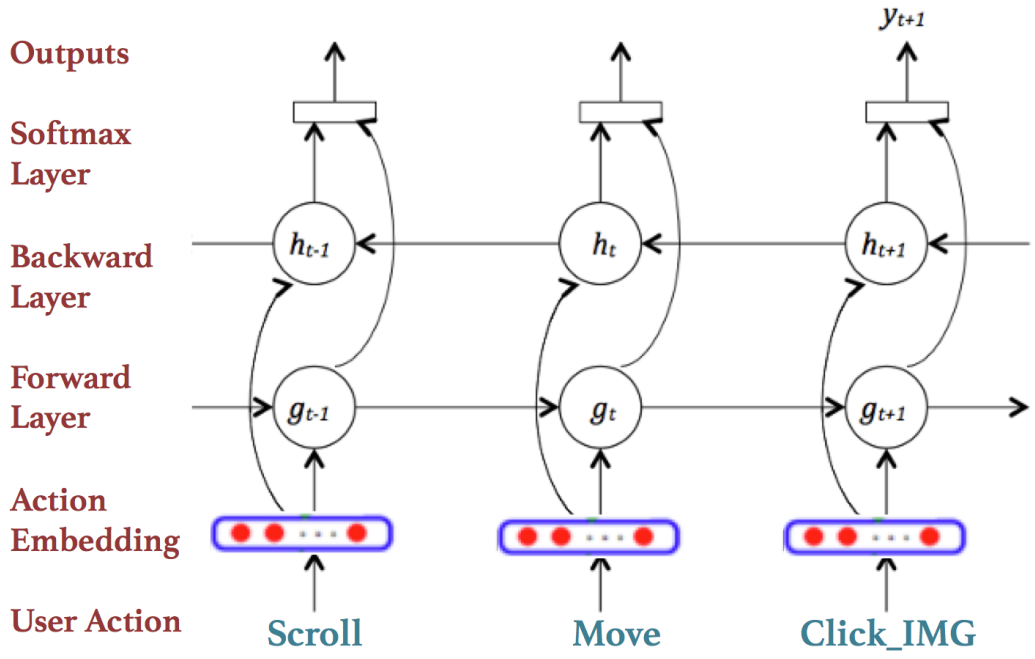


Figure 8.2: The Bi-directional LSTM model for query SAT prediction.

$y = (y_1, y_2, ..., y_T)$ by iterating from $t = 1$ to T:

$$h_t = \mathscr{H}(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h) \tag{8.2}$$

$$y_t = \mathbf{W}_{hy}h_t + b_y \tag{8.3}$$

where T is the total number of sequences; $\mathbf{W_{xh}}$ are the weight matrices between the input layers a and h and so on; b is a bias vector, and $\mathscr{H}$ is the composite function. The action-LSTM architecture is composed of two components: (i) Action Embeddings and (ii) LSTM sequence model. We next discuss both these components in detail.

Action Embeddings:

The input to the action-LSTM is the sequence of user actions on the rendered SERP. While one-hot vector representations have been traditionally used as input to the recurrent neural networks, recently embeddings have shown enhanced performance. We learn action embeddings from the interaction sequence data. Given the set of action sequences, the first layer embeds each action into a continuous

vector space using a skip-gram model [122]. Since the input sequences are of arbitrary length, we mask the input sequences with dummy symbol which are ignored during training phase. The embedding layer is optimized jointly with the rest of the model through backpropagation, [189] optimizing the individual actions' embedding vectors to be more reflective of their semantic closeness to other actions.

Sequence LSTM Model:

After passing through the embedding layer, the input action sequences are input to the LSTM module. The LSTM composite function forming the LSTM cell with peephole connections is defined as:

$$\mathbf{i_t} = \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}h_{t-1}) \tag{8.4}$$

$$\mathbf{f_t} = \sigma(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}h_{t-1}) \tag{8.5}$$

$$\mathbf{c_t} = \mathbf{f_t} \odot \mathbf{c}_{t-1} + i_t \odot tanh(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}h_{t-1}) \tag{8.6}$$

$$\mathbf{o_t} = \sigma(\mathbf{W}_{hx}x_t + \mathbf{W}_{ho}h_{t-1} + \mathbf{W}_{co}\mathbf{c}_t) \tag{8.7}$$

$$\mathbf{h_t} = \mathbf{o_t} \odot tanh(\mathbf{c}_t) \tag{8.8}$$

where σ denotes the sigmoid function, $\sigma(z) = (1+e^{-z})^{-1}$. The superscripts $^{(t)}$ denote the index of the current time step, **i**, **f** and **o**, are respectively the input, forget and output gates, and **c** the cell activation vector with the same size than the hidden vector **h**. The weight matrices **W** from cell **c** to gates **i**, **f** and **o**, are diagonal, and thus, an element e in each gate vector receives only the element e from the cell vector.

In any action in a interaction sequence, we not only have historic actions, but also have future actions user took on the SERP. For many sequence labelling tasks it is beneficial to have access to both past (left) and future (right) actions contexts. However, the LSTM's hidden state h_t takes information only from past, knowing nothing about the future. To leverage future action information, we use bi-directional LSTM (BLSTM) wherein the basic idea is to present each sequence forwards and backwards to two separate hidden states to capture past and future

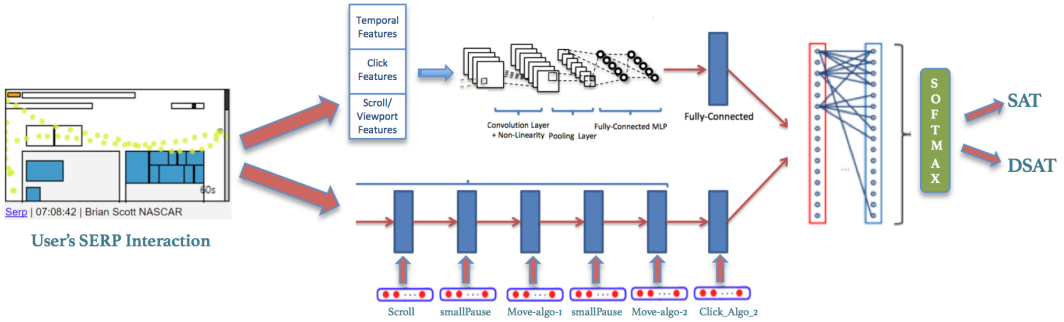


Figure 8.3: Neural architecture of the proposed deep Unified Multi-view CNN-LSTM model.

information, respectively. It is important to note that our goal is retrospective satisfaction prediction, i.e., offline prediction of user satisfaction based on the observed interaction signals. While future action sequences will not be available in an online setting, this restriction does not apply in our offline setting, as a result, bi-directional LSTMs can be used in retrospective offline satisfaction prediction. This type of RNN feeds to a same output layer fed forwarded inputs through the two hidden layers. Therefore, the BLSTM computes both forward hidden sequence $\vec{h}$ and backward sequence $\overleftarrow{h}$ as well as the output vector y, by iterating $\vec{h}$ from t = 1 to T , and $\overleftarrow{h}$ from t = T to 1:

$$\vec{h}_t = \mathscr{H}(\mathbf{W}_{x\vec{h}}x_t + \mathbf{W}_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \tag{8.9}$$

$$\overleftarrow{h}_t = \mathscr{H}(\mathbf{W}_{x\overleftarrow{h}}x_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \tag{8.10}$$

$$y_t = \mathbf{W}_{\vec{h}y}\vec{h}_t + \mathbf{W}_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \tag{8.11}$$

where $\mathscr{H}$ is the composite function. The BLSTM allows to exhibit long range context dependencies and takes advantage from the two directions structure. The output vector y is processed by evaluating simultaneously the two directions hidden sequences by computing the composite function H in the forward and backward directions.

8.4.2 Unified Multi-View Interaction Model

Although sequence based approaches to satisfaction prediction are an effective way of capturing user interactions, we hypothesize that better, richer representation of

user activity can be obtained by incorporating other interaction signals in the model architecture. The traditionally used static features and implicit signals provide a different view of the user interactions. Our primary contribution here is a novel neural architecture that is designed to jointly leverage sequence information with such static implicit feedback signals to predict search satisfaction.

To illustrate, consider the example of a user action sequence: Pause – Scroll – Click. While sequence information is informative, aggregate metrics such as dwell times etc provide useful cooked information and is helpful in capturing domain information about user behavior with SERP.

8.4.2.1 Auxiliary Signals

A number of different interaction behaviors have been taken into consideration in the prediction of search user satisfactions including both coarse-grained features (e.g. clickthrough based features [99]) and fine-grained ones (e.g. cursor position and scrolling speed [100]). We use a number of such traditionally used signals as auxiliary side-information which provides an alternative view of user interaction. We categorize these signals into three groups: (i) Temporal signals, (ii) Click based signals and (iii) Scroll & pointer signals. Table 8.3 presents the different types of signals captured under each of these groups which provide us an alternative view of the user interaction. We next describe our model which jointly encodes these auxiliary features with the sequential action-LSTM model.

8.4.2.2 Unified Multi-View Interaction Model

The auxiliary signals described above provide us with an alternate view of user interaction. We use these auxiliary signals to enrich our sequential model to create a unified multi-view model of user interactions. We propose a coupled architecture composed of deep convolutional network and dense layers for modelling auxiliary features and couple it with the action-LSTM architecture described before. Its main building blocks are (i) action-LSTM which use the action sequences and (ii) the auxiliary feature module based on convolutaional neural networks (ConvNets), both of which work in parallel mapping details of user interactions to their distributional vectors which are then used to predict user satisfaction for each query.

Feature Set	Feature List
Temporal Signals	Page dwell time
	Reading time per pixel
	Viewport time per instance
	Time to first pointer event
	Time to first scroll event
Click based Signals	Total click count
	Algo click count
	Answer click count
Scroll & Pointer Signals	Total scroll count
	Pointer horizontal distance
	Pointer vertical distance
	Pointer event count
	Scroll Up count
	Scroll down count
	Viewport direction changes

Table 8.3: Auxiliary Signals: List of implicit signals used as side information.

The architecture of our ConvNet for mapping implicit signals to features is mainly inspired by the various CNN architectures used for performing different classification tasks. However, different from previous work the goal of our distributional auxiliary signals model is to learn good intermediate representations of such signals, which are then coupled together with the output representation of the sequential action-LSTM model and used for satisfaction prediction. The input to the ConvNet module are the three set of implicit feedback signals (as shown in Table 8.3) that are processed by intermediate convolutional layers. The aim of the convolutional layers is to extract patterns, i.e., discriminative signal sequences found within the input signals that are common throughout the training instances.

More specifically, the convolution operator operates on sliding windows of signals, and the convolutions in deeper layers are defined in a similar way. Suppose we have a discrete input function $g(x) \in [1,l] \rightarrow R$ and a discrete kernel function $f(x) \in [1,k] \rightarrow R$. The convolution $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow R$ between f(x) and g(x) with stride d is defined as:

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c) \tag{8.12}$$

where c = k - d + 1 is an offset constant. The module is parameterized by a set of such kernel functions $f_{ij}(x)(i = 1,2,...,m$ and $j = 1,2,...,n)$ which we call weights, on a set of inputs $g_i(x)$ and outputs $h_j(y)$. The output from the convolutional layer (passed through the activation function) are then passed to the pooling layer, whose goal is to aggregate the information from the previous layer. Given a discrete input function $g(x) \in [1,l] \rightarrow R$, we employ a 1-D spatial max-pooling function $h(y) \in [1, \lfloor (l-k)/d \rfloor + 1] \rightarrow R$ of g(x) defined as:

$$h(y) = max_{x=1}^{k} g(y \cdot d - x + c) \tag{8.13}$$

where c = k - d + 1 is an offset constant. To allow the network learn non-linear decision boundaries, each convolutional layer is typically followed by a non-linear activation function applied element-wise to the output of the preceding layer. The non-linearity used in our model is the rectifier or thresholding function

$$h(x) = max\{0,x\} \tag{8.14}$$

which makes our convolutional layers similar to rectified linear units (ReLUs) [190].

Convolutional layer passed through the activation function together with pooling layer acts as a non-linear feature extractor. Finally, we combine the ConvNet feature extractor with the output of the action-LSTM and pass it through 2 dense layers. and a softmax layer at the end.

Interaction Layers:

Our model includes an additional hidden layer right before the softmax layer (described next) to allow for modelling interactions between the components of the intermediate representation, i.e., the different views of user interactions. The hidden layer computes the following transformation: $\alpha(w_h \cdot x + b)$ where w_h is the weight vector of the hidden layer and $\alpha()$ is the ReLU non-linearity function.

Softmax Layer:

The output of the penultimate convolutional and pooling layers is flattened to a dense vector x, which is passed to a fully connected softmax layer. It computes the probability distribution over the labels:

$$p(y = j|x) = \frac{e^{x^T \theta_j}}{\sum_{k=1}^{K} e^{x^T \theta_k}} \tag{8.15}$$

where θ_k is a weight vector of the k-th class. x can be thought of as a final abstract representation of the input example obtained by a series of transformations from the input layer through a series of convolutional and pooling operations.

8.4.3 Training

The parallel multi-view CNN-LSTM model is trained to minimize the RMSE error on satisfaction prediction accuracies. We use the ADAM optimization algorithm for training [191], with a batch size of 64. The learning rate is initially chosen as 0.01, and dropped to 0.003 in the middle of training before convergence. We useed the standard default values for other parameters of the optimizer: β_1 = 0.9, β_2 = 0.999, $\varepsilon = 10^8$. While neural networks have a large capacity to learn complex decision functions they tend to easily overfit especially on small and medium sized datasets. To mitigate the overfitting issue we insert 2 dropout modules in between the fully-connected layers to regularize. They have dropout probability of 0.2. Dropout prevents feature co-adaptation by setting to zero (dropping out) a portion of hidden units during the forward phase when computing the activations at the softmax output layer and also acts as an approximate model averaging [192].

8.5 Functional Composition for Task Satisfaction

Given details of user interactions at the query level and the corresponding query level satisfaction prediction architecture proposed in the previous section, our overall goal is to make task satisfaction predictions. In this section, we enumerate different ways of using query satisfaction predictions to make task level satisfaction predictions. Specifically, given a sequence of queries $Q = q_1, q_2, ..., q_t$ belonging to

a search task $t \in T$, where T is the set of all tasks, the Multi-view CNN-LSTM architecture provides us with estimates of query level satisfaction $Y_{q_i} = \varphi_q(q_i, a_{q_i})$ where $Y_{q_i} \in \{0,1\}$ is the query level satisfaction estimate, a_q is the set of action sequence observed for the search impression for query q and φ_q is the query level satisfaction prediction function. Our goal is to make task level satisfaction prediction:

$$y_t = F\Big(\Big\{q_1, \varphi_q(q_1)\Big\}, \Big\{q_2, \varphi_q(q_2)\Big\}, \Big\{q_3, \varphi_q(q_3)\Big\}, ..., \Big\{q_i, \varphi_q(q_t)\Big\}\Big) \tag{8.16}$$

where **F**: $\{q_1, \varphi_q(q_1)\} \rightarrow Y_t \in 0,1$ represents the functional transformation which maps query-satisfaction estimate tuple $\{q_i, \varphi_q(q_i)\}$ to a task satisfaction label. Based on known insights on task satisfaction, we present a number of different functional compositions techniques at two levels of abstract: (i) query level aggregation and (ii) subtask level aggregation.

8.5.1 Query level composition

To make task level satisfaction predictions, we being by aggregating satisfaction signals at the query level. We consider four distinct functional forms of aggregation, ranging from extremely strict to lenient evaluation of task satisfaction.

1. **Maximum**: This functional composition method assumes that the user is satisfied in completing their task if they are satisfied in any of the queries they issued while completing the task. Specifically:

$$y_t = max\Big(\varphi_q(q_1), \varphi_q(q_2), \varphi_q(q_3), ..., \varphi_q(q_t)\Big) \tag{8.17}$$

   where $\varphi_q(q_i)$ gives the query level satisfaction estimate based on the Multi-View CNN-LSTM architecture. It is to be noted that such a functional composition is the most lenient way of evaluating search engine performance.

2. **Average**: This functional composition techniques considers equal contribution from each query in predicting task satisfaction. Specifically, $y_t = \frac{\sum_{i=1}^{|Q_t|} \varphi_q(q_i)}{|Q_t|}$ where $|Q_t|$ is the number of queries associated with the task t.

3. **Differential Weighting:** Often users reformulate their information needs and

issue a series of queries as they complete their task. We hypothesize that queries towards the end of the task are more important than the ones at the start, based on which we over emphasize queries towards the end of the task when considering their contribution towards task satisfaction. Specifically:

$$y_t = \frac{\sum_{i=1}^{|Q_t|} w_i \varphi_q(q_i)}{|Q_t|} \tag{8.18}$$

where w_i is the weight associated with query q_i.

4. **Minimum**: This functional composition assumes that a user is satisfied is completing their task if they are satisfied in each of the queries they issued to accomplish the task. Specifically:

$$y_t = min\Big(\varphi_q(q_1), \varphi_q(q_2), \varphi_q(q_3), ..., \varphi_q(q_t)\Big) \tag{8.19}$$

Such an computation of task satisfaction is the most strict estimate of task satisfaction since if any SERP rendered for query is unsatisfying to the user, the whole task is rendered unsatisfying.

8.5.2 Subtask based composition

Often search tasks involve many distinct, but related aspects which warrant the need for issuing different sets of queries over time in order to fulfill the multi-aspect information needs. A complex search task could be broken down into smaller multi-step or multi-aspect sub-tasks that represent atomic informational needs, for which it is trivial for users to issue satisfying queries. We hypothesize that task-level satisfaction could estimated based on user's satisfaction levels when attempting different subtasks. An ideal task completion engine would help the user satisfactorily accomplish each of the associated subtasks. We utilize this insight to estimate task satisfaction from the associated subtask satisfaction estimates.

Given a task t composed of $|S_t|$ subtasks, we consider a nested functional composition of satisfaction estimates at two levels: (i) aggregating query satisfaction estimates to compute subtask satisfaction and (ii) aggregating subtask satisfaction

Method Type	Method	Accuracy	Precision	Recall	FMeasure	Log-Loss
Feature based baselines	Baseline 1(Clicks+Dwell Time)	0.561	**0.86**	0.58	0.6927	13.88
	Baseline 2 (Click based actions)	0.593	0.78	0.61	0.6846	13.67
	Baseline 3 (Mouse Movement)	0.606	0.72	0.66	0.6886	13.32
	Baseline 4 (Scroll & Viewport))	0.586	0.71	0.67	0.6894	13.73
	Baseline 5 (Reading Pattern Signals)	0.596	0.72	0.69	0.7046	13.61
Sequential baselines	Generative Probabilistic	0.631	0.81	0.67	0.7333	13.04
	CRF-Actions	0.593	0.77	0.6	0.6744	14.74
	CRF-Queries	0.582	0.75	0.62	0.6788	14.89
Proposed/Variants	SimpleRNN	0.654	0.72	0.85	0.7796	11.36
	action-Embedding + LSTM	0.668	0.71	0.88	0.7859	11.08
	action-Embedding + Bi-LSTM	**0.677**$^{*\&}$	0.73	**0.89**$^{*\&}$	**0.8020**$^{*\&}$	**10.98**$^{*\&}$

Table 8.4: Query level SAT prediction. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the best performing feature based baseline and the best performing sequential baseline respectively.

estimates to compute task satisfaction. Specifically,

$$y_t = f\Big(g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_1\Big), g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_2\Big), \\ ..., g\Big(\{\varphi_q(q_i)\}\forall q_i \in S_t\Big)\Big)$$

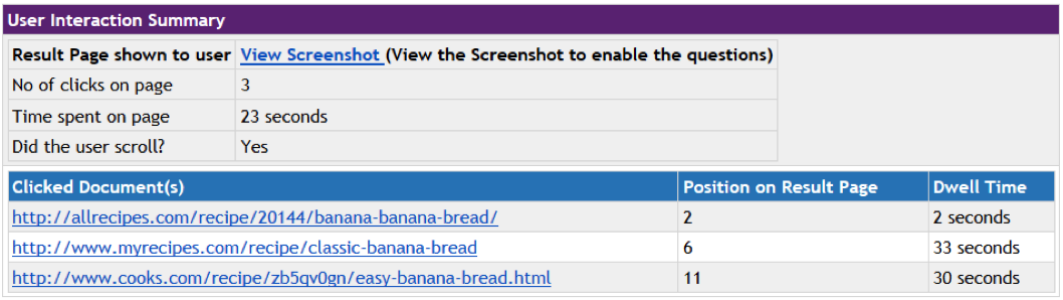
where S_i represents the subtask j and S_t represents the total number of subtasks in the task t. The functions f(.) and g(.) could be either of the four query level aggregate functions defined before. While there exist automated subtask extraction approaches [186], for the scope of this chapter, we assume access to subtask demarcation information obtained via crowdsourced labeling.

8.6 Experimental Evaluation

In this section, we demonstrate how our satisfaction prediction models perform for predicting both query and task level satisfaction. We conduct a number of experiments using crowdsourced judgments as well as real world search engine traffic. We make use of labels obtained via crowdsourced judgments studies as ground truth labels for all evaluations considered; however, we leverage large scale pseudo-labelled data with weak supervision signals to train our deep models.

8.6.1 Dataset

Our data consists of a random sample of user sessions from a major US commercial search engine engine during a week in June 2016. We randomly sampled user



User Interaction Summary	
Result Page shown to user	View Screenshot (View the Screenshot to enable the questions)
No of clicks on page	3
Time spent on page	23 seconds
Did the user scroll?	Yes

Clicked Document(s)	Position on Result Page	Dwell Time
http://allrecipes.com/recipe/20144/banana-banana-bread/	2	2 seconds
http://www.myrecipes.com/recipe/classic-banana-bread	6	33 seconds
http://www.cooks.com/recipe/zb5qv0gn/easy-banana-bread.html	11	30 seconds

Figure 8.4: Summary of user interaction on the SERP shown to judges.

sessions with substantial user activity, and included all queries, search result page impressions on all results on the search result page from that user in the timeframe. Additionally, detailed user activity on the result page was logged for model development. In total, our sample contained No of sessions over 14670 search sessions, resulting in about 148561 search queries.

8.6.1.1 Large Scale Pseudo-Labelled Data

While we collect crowdsourced labels for creating ground truth labels, owing to the limited scale of experimentation possible with crowd-sourced judgments as well as the differences in opinion of crowdsourced judges and actual users, we may have insufficient data and labels to reliably train deep parameter-rich models. To resolve this problem, we build a pseudo-labeled dataset comprised of the entire large-scale query log described in Section 8.6.1. To assign pseudo satisfaction labels to search interactions, we assume that a click followed by a query reformulation is a dissatisfied click, while a click with a dwell time of ≥ 30 seconds not followed by a query reformulation is a satisfied click. Post-click query reformulation is considered a strong DSAT predictor and has been used as a predictor of search satisfaction in previous work [119, 188]. To identify query reformulations we use a method similar to that described in Boldi *et al.*[193], where features of query similarity (e.g. edit distance, word overlap, etc.) and time between queries are used to identify query reformulations. To remain conservative while creating the pseudo-labelled data, we ignored all other cases and only considered the cases highlighted above wherein the system was sure of a satisfactory or unsatisfactory experience.

8.6.2 Collecting Task SAT Judgements

Crowdsourced judgments have commonly been used to obtain labeled data [194, 195]. To gauge user satisfaction at both query level and task level, we collect judgement labels at both levels. For each search impression as well as the overall task, we obtained human labelled judgments on whether the user interaction was satisfying (labelled SAT) or not (labelled DSAT). The labelling was conducted using an in-house microtasking platform that outsources crowd work to vendors, similar to CrowdFlower, and provides access to judges who regularly perform relevance judgment tasks. Workers were under NDA and all data containing personal identifiable information (PII), such as names, phone numbers, addresses, or social security numbers, were removed.

Detailed guidelines were issued to the judges to describe the task and a number of examples were shown defining what constitutes a query, a subtask and a task and explaining how to judge for query as well as task level satisfaction. To ensure the quality of the judging results, we apply a series of quality control methods. One of the methods is creating 'gold hits' that you already know the answer of, then measure the judges by comparing how far off their answers are from the gold hits answers. We also measure the quality of the judgments with the amount of consensus reached which required overlap on the hits, i.e. the same hit to be judged by multiple judges.

The data presented to the judges come from previously annotated data where another group of judges defined the task boundaries within a session. In other words, each session was divided into one of more coherent tasks. A sequence of queries are considered part of a coherent task if they collectively try to achieve a certain goal. The output of the task boundary annotation is given to our group of judges where each is represented as a series of queries along with the corresponding user interaction information. In order to provide relevant information to the judges, we provided a detailed summary of user interaction with the SERP. The judges were provided a link to the SERP shown to the user alongside details like number of clicks, time spent on the SERP and scroll information. Additionally, for all the

Method	Accuracy	Precision	Recall	FMeasure	Log-Loss
CRF-Actions	0.593	0.77	0.6	0.6744	14.74
CRF-Actions + All Signals	0.603	0.78	0.61	0.6848	14.29
Generative Probabilistic	0.631	0.81	0.67	0.7333	13.04
Generative Probabilistic + All Signals	0.651	0.823	0.681	0.744	12.97
action-Embedding+ LSTM	0.677	0.73	0.89	0.8020	10.98
action-Embedding+ LSTM + Click based Signals	0.678	0.744	0.825	0.783	11.11
action-Embedding+ LSTM + Temporal Signals	0.699	0.714	**0.954**$^{*\&}$	**0.817**$^{*\&}$	10.36
action-Embedding+ LSTM + Scroll/Viewport Signals	0.689	**0.728**$^{*\&}$	0.89	0.801	10.72
Unified Multi-View Model (action-LSTM + All Signals)	**0.703**$^{*\&}$	0.717	0.944	0.815	**10.25**$^{*\&}$

Table 8.5: Evaluating the unified model for Query SAT prediction. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF all signals and Generative Probabilistic - All Signals baselines respectively.

clicked documents, we provided URL level details which included the exact URL, the position on the SERP where it was shown and the total dwell time on each URL. Each judge was asked to consider the user interaction summary and provide labels for query and task satisfaction. Figure 8.4 provides an example summary of user interaction.

We randomly sampled over 2100 user tasks and over 450 judges provided judgments for about 6820 search impressions, resulting in over 20460 judgments. Among the first two judgments collected for each query, the judges agreed on the label 74% of the time. We measured inter-rater agreement using Fleiss' Kappa [196], which allows for any number of raters and for different raters rating different items. This makes it an appropriate measure of inter-rater agreement in our study since different judges provided labels for different items. A kappa value of 0 implies that any rater agreement is due to chance, whereas a kappa value of 1 implies perfect agreement. In our data, $\kappa = 0.64$, which, according to Landis and Locke [197], represents substantial agreement.

8.6.3 Baselines

We consider a number of baselines from recent published literature, including both non-neural and neural models, as well as non-sequential and sequence based models.

- **Baseline 1 (click with dwell time):** Spending a minimum amount of time on a webpage is known as a long dwell click and has been shown to be correlated with satisfaction [188]. In this study, we set t = 30 seconds.

- **Baseline 2 (click based actions):** This baseline is based on predicting satisfaction based on clickthrough based features [100].
- **Baseline 3 (Mouse movement):** This baseline is based on recent work aimed at predicting satisfaction using mouse movement patterns [114].
- **Baseline 4 (Scroll & Viewport):** This baseline is based on the recently proposed scrolling and viewport features [108, 195]
- **Baseline 5 (Reading pattern signals):** This baseline is based on the reading pattern signals from Kiseleva *et al.*[198]

Additionally, we consider a number of sequence based models to compare the performance of the proposed approach.

- **Generative Probabilistic Model:[199]** A semi-supervised generative model wherein every action sequence is generated using a probability distribution specified by a 2-component mixture model.
- **CRF Models:** Conditional random field models are popularly used for many different sequence labeling tasks. We consider two variants of CRF models based on the input features they use:
  - action-CRF: this CRF makes use of only the action information for constructing CRF features.
  - query-CRF Model: in addition to action co-occurrence features, this CRF model takes into account query level features during training.

We also consider variants of the proposed model: (i) simple RNNs, (ii) action-embedding LSTM and (iii) action-Embedding Bi-LSTM.

8.6.4 Query Level SAT Prediction

As our first experiment, we consider predicting user satisfaction for each search impression. We compare the proposed sequential action embedding + LSTM model with traditionally used features as well as other popular feature based and sequential models. For each query, we extract the set of features needed by the different

Functional Composition Type	Method	Accuracy	Precision	Recall	FMeasure	Log-Loss
Maximum	CRF	0.639	0.914	0.629	0.7454	12.46
	Generative Probabilistic	0.811	**0.917**	0.852	0.883	6.49
	action-Embedding Bi-LSTM	0.823	0.841	0.974	0.902	6.08
	Unified Multi-View	**0.8309**$^{*\&}$	0.838	**0.988**$^{*\&}$	**0.907**$^{*\&}$	**5.83**$^{*\&}$
Minimum	CRF	0.5259	0.785	0.589	0.679	16.37
	Generative Probabilistic	**0.826**	0.838	**0.983**	**0.904**	**6.003**
	action-Embedding Bi-LSTM	0.618	**0.94**$^{*\&}$	0.356	0.517	19.32
	Unified Multi-View	0.5952	0.882	0.597	0.712	13.98
Average	CRF	0.57	0.906	0.544	0.6807	14.82
	Generative Probabilistic	0.797	**0.918**	0.832	**0.873**	6.99
	action-Embedding Bi-LSTM	0.6809	0.847	0.756	0.799	11.
	Unified Multi-View	**0.801**$^{*\&}$	0.842	**0.895**$^{*\&}$	0.868	**6.89**$^{*\&}$
Differentially Weighted	CRF	0.632	0.913	0.62	0.739	12.7
	Generative Probabilistic	0.814	**0.917**	0.855	0.885	6.41
	action-Embedding Bi-LSTM	0.714	0.853	0.781	0.815	8.21
	Unified Multi-View	**0.824**$^{*\&}$	0.849	**0.929**$^{*\&}$	**0.887***	**6.84**$^{*\&}$
Subtask (Max-Average)	CRF	0.591	**0.926**	0.553	0.6924	13.66
	Generative Probabilistic	0.761	0.901	0.812	0.8541	8.89
	action-Embedding Bi-LSTM	0.70	0.83	0.79	0.82	10.36
	Unified Multi-View	**0.77**$^{*\&}$	0.84	**0.89**$^{*\&}$	**0.86***	**8.14**$^{*\&}$
Subtask (Average-Max)	CRF	0.621	0.904	0.632	0.7439	12.89
	Generative Probabilistic	0.814	**0.921**	0.841	0.8791	6.41
	action-Embedding Bi-LSTM	0.79	0.85	0.92	0.88	7.56
	Unified Multi-View	**0.838**$^{*\&}$	0.84	**0.98**$^{*\&}$	**0.91**$^{*\&}$	**6.08**$^{*\&}$

Table 8.6: Task level SAT prediction. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF and Generative Probabilistic baselines respectively.

baselines as well as the detailed user interaction action sequence and consider the judgment labels obtained from the crowdsourced study as the ground truth. We randomly split the data into training and test set in 60/40 ratio. We use the pseudo-labelled data described in Section 8.6.1.1 to pre-train the neural models.

Table 8.4 presents the prediction results comparing the proposed approach with established baselines. We observe that sequence based baselines perform better than feature based baselines in general, with the generative probabilistic baseline performing particularly better with over 7% improvement in accuracy scores. A satisfying click, i.e., click followed by a long dwell time, has traditionally been used to gauge user satisfaction. We re-confirm such known insights since we observe that Click+DwellTime obtain the best precision; however this method misses out on capturing various other satisfactory interactions, as is evident from their low recall scores. Further, we observe that mouse movement information (baseline 3) in general is more predictive than just click based features.

Overall, we observe that the proposed deep sequential model and its variants outperform all baselines considered in predicting user satisfaction and register an

improvement in over 11% over the worst performing baseline and ∼5% over the best performing generative sequence modelling approach. Among the variants considered, the simple RNN model is outperformed by the more sophisticated LSTM models which confirms known benefits offered by LSTMs over RNNs. The bidirectional version of the proposed model outperforms the LSTM model on all metrics, which confirms our hypothesis that including future action signal information helps in modeling user interaction better. Indeed, since most satisfaction detection and evaluation is performed post-hoc, and historic data logs entire user interactions, future actions signal information is readily available and should be used in modeling user interactions. The proposed deep sequential models perform significantly better in terms of recall, with obtaining 20% improvement over the best performing baselines. This strongly suggests that the rich user interaction signals used by our deep sequence models are perhaps able to capture and detect user satisfaction in non-click scenarios, and abandonment cases.

8.6.5 Unified View for QSAT

We next evaluate the benefit of unifying the different interactions signals, both static features and interaction sequences. We investigate how adding different sets of features to the sequential model help in better predicting user satisfaction. Table 8.5 presents the results on query level satisfaction prediction comparing the proposed Unified Multi-View CNN+LSTM model with the best performing baselines.

We observe that adding the other view of user interaction data always helps in improving prediction performance across all methods. Adding click based signal information to the interaction sequence information improves SAT precision (at the cost of recall), which is consistent with what was observed before. Adding temporal signals give a significantly improved performance in terms of recall, with over 27% improvement in detecting satisfaction cases which may have otherwise been missed by baseline approaches. Indeed, temporal signals and detailed user interactions go well beyond shallow methods which assume a very restrictive view of user satisfaction. Further, the unified multi-view model achieves the best accuracy in predicting user satisfaction with over 5% improvement in accuracy, 26% improve-

Method	Accuracy	Precision	Recall	FMeasure	Log-Loss
CRF	0.639	**0.914**	0.629	0.7454	12.46
Generative Probabilistic	0.826	0.838	0.983	0.9045	6.003
action-Embedding Bi-LSTM	0.823	0.841	0.974	0.9022	6.08
Unified Multi-View	**0.843**$^{*\&}$	0.851	**0.991**$^{*\&}$	**0.9156**$^{*\&}$	**5.62** $^{*\&}$

Table 8.7: Comparing the performance of different task SAT prediction approaches across all functional compositional techniques. * and & indicate statistical significant ($p \leq 0.05$) using paired t-tests compared to the CRF and Generative Probabilistic baselines correspondingly.

ment in recall and 7% improvement in F-score. These results strongly demonstrate the benefits offered by the enriched unified multi-view models by leveraging not only the interaction sequence information, but also other static implicit signals.

8.6.6 Task SAT Prediction

One major motivation for the current work is to leverage user interaction signals to predict task level satisfaction of users. To this end, we consider the problem of task satisfaction prediction and compare how the different compositional functions perform in predicting task level satisfaction. Since we collected task satisfaction judgements alongside query level satisfaction judgements, we make use of these task level judgements as ground truth information.

Before diving deep into different compositional functions, we first look at how the proposed models perform on the task satisfaction problem. As shown in Table 8.7, we observe that the proposed deep sequential model performs better than the best performing baselines in predicting task satisfaction across all five metrics. Moreover, the unified multi-view model performs better than the deep sequential model, which demonstrates that the combined information from interaction sequences and other auxiliary implicit feedback signals are not only good for query level satisfaction prediction, but also work best at measuring task satisfaction.

We additionally analyze how the different functional composition techniques fare. Table 8.6 presents the task satisfaction prediction results wherein we compare the proposed models with best performing baselines across the different functional composition techniques. We considered five different functional composition techniques for aggregating query level satisfaction estimates to compute task satis-

faction. We observe that the most lenient aggregating technique (*Maximum*) consistently achieves higher accuracy than the most strict satisfaction criterion (*Minimum*). We observe that the differential weighting scheme performs better than the average function, which hints at the fact that not all queries contribute the same towards a task. Finally, considering subtasks information in the intermediary stage between query and task level abstractions helps in better predicting task satisfaction.

8.7 Conclusion

We considered a holistic view of user interaction and presented deep sequential models for predicting user satisfaction at various levels of abstraction. While most exiting approaches focus on query satisfaction or task satisfaction for simple atomic tasks, we go beyond such atomic tasks and consider the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks. The proposed unified multi-view model and the functional composition approach performs better than a number of established baselines. We hope that the findings of this work would inspire future research in developing sophisticated techniques for quantifying the importance of different queries and subtasks in any given complex task. For instance, certain queries would be more important to accomplish a task, and certain subtasks could be option and be potentially skipped. Further, task satisfaction prediction could inspire research in developing retrieval algorithms optimized for task completion. Finally, we contend that the promising results demonstrated by the unified multi-view approach would help in improving satisfaction prediction and good abandonment detection on mobile devices.

Chapter 9

Conclusion & Future Work

In this thesis, we have investigated user's task behavior and presented techniques for understanding, extracting and leveraging search tasks. We have done so in two parts: in Part I we have studied user behavior and search tasks, presented techniques for extracting tasks-subtasks, and in Part II we have leveraged task information to learn user models, query representations and gauge user satisfaction.

Part I revolved around understanding and extracting search task. Recent advances in understanding online search behavior has introduced the idea of search tasks as the atomic unit of search activity on the web. We illustrate in this paper how a shift of focus from the idea of a search session to a search task raises a number of important questions. First in chapter 3, we investigated behavioral insights into how users interact with search systems and accomplish various search tasks. We presented analysis on user's propensity to multi-task, user groups based on multi-tasking habits of users, and the interplay between user-disposition, topic-level heterogeneities and search tasks. In order to fully understand the different aspects of complex search tasks, we presented a bayesian non-parametric subtask extraction algorithm in Chapter 4 which extracted subtasks from a given collection of *on-task* queries without specifying the expected number of subtasks apriori. In chapter 5, we showed that a more naturalistic view of looking at tasks consists if hierarchies of tasks with complex tasks hierarchically decomposed into more focussed subtasks. We presented a hierarchical bayesian non-parametric model to extract these hierarchies and presented a comprehensive evaluation setup to quantify the quality of the

extracted hierarchy.

In Part II, we turned to different ways in which the task information could be useful in helping search systems better serve users. In chapter 6, we presented an application of leveraging task information and showed that incorporating task information indeed helps in developing better user models for personalization. Beyond user representations, we also showed that task information helps in learning better query representations. Specifically, in Chapter 7 we presented a novel task based embedding architecture that learns distributed semantic representation of query terms preferring task context over local information in immediately surrounding words. Finally, moving beyond user and query representations, in Chapter 8, we took a comprehensive look at user satisfaction from different levels of abstractions. We proposed a deep sequential model which considers holistic view of user interaction with the search engine result page (SERP), constructs detailed interaction sequences of their activity and leverages such interaction sequences to predict query level satisfaction. In this chapter, we also proposed a novel functional composition model which takes into account user satisfaction at the query as well as the subtask level to make task level satisfaction predictions.

In this concluding chapter we first look back at the research questions we sought to answer in the first chapter of this thesis, in Section 1.1, and summarize the answers to our research questions asked in Parts I and II. We briefly summarize all our findings once more in Section 10.2, discuss implications of our findings in Section 10.3 and present some limitations of the work in Section 10.4. Building on top the presented limitations, we reflect on few promising future research directions in Section 10.6.

9.1 Main Findings

Here we summarize in detail the methods and findings presented in this thesis in line with the Research Questions raised in Chapter 1.

Part I: Understanding and Extracting Tasks

An important starting question we investigated dealt with fully characterizing the extent and underlying heterogeneities surrounding single-task and multi-task search sessions. The answers to these questions would be instrumental for search engine platforms to generate more accurate personalization strategies for their users. In Chapter 3, we emphasized that while most users on search engines are multi-task users performing 2 or more tasks within a single search session, there exist a sizeable proportion of users who are more focussed and mostly mono-task. We also provide evidence of "Supertaskers" who perform onwards of 4 tasks within a single session. As highlighted by the varying search effort metrics, the different user groups indeed interact with the search results differently and hence motivate the need for incorporating such differences in multi-tasking behavior of users while personalizing search experiences. Yet another finding we wish to highlight through our study is the characterizing of multiple tasks into a combination of a single primary and multiple ancillary tasks. Our task effort scores provide preliminary evidence to suggest that such categorization of multiple tasks into a task-hierarchy might indeed be plausible.

While we draw on previous work as well as our own set of analyses to show that multi-tasking within a search session is fairly common, we also emphasize that the extent and nature of multi-tasking is strongly influenced by user dispositions (i.e. whether a user is naturally disposed to single vs. multi-tasking), topic preferences (i.e. users might prefer to multi-task when searching for certain topics than others), and interest preferences (i.e. users might prefer to multi-task about topics they are more or less interested in).

Prior work on identifying search-tasks mainly explores task extraction from search sessions with the objective of segmenting a search session into disjoint sets of queries where each set represents a different task. Despite these efforts, problems with existing task extraction systems abound. We presented two novel algorithms for task and subtask extraction and experimentally demonstrated that the proposed task extraction algorithms are able to extract richer set of tasks and subtasks than

existing baselines. In Chapter 4, we exploited the concept distributional semantics with non-parametric priors and presented a bayesian non-parametric model for extracting coherent subtasks from a given query collection of *on-task* queries. The proposed generative model (TE-coh-ddCRP) is not restricted by a fixed number of sub-task clusters, and assumes an infinite number of latent groups, with each group being described by a certain set of parameters. We specified our non-parametric model by defining a Distance-dependent Chinese Restaurant Process (dd-CRP) prior and a Dirichlet multinomial likelihood and enriched the model by working in the vector embedding space which uses a word-embedding based distance measure to encode query distances for efficient sub-task extraction. Further, we formally defined the notion of subtask affinity, which helps us quantify the semantic cohesiveness and coherence of a given subtask, based on which we propose a novel likelihood function which encodes the coherence estimates.

Based on several experiments presented in Chapter 4, we demonstrated that the proposed model indeed helps in extracting coherent subtasks from a collection of *on-task* queries. Using a user judgement study, we measured the subtask relatedness score which provides an estimate of how coherent the extracted subtasks are and showed that the proposed method outperforms all the baselines considered, in making correct sub-task assignments. Further, we demonstrated the efficacy of the model by measuring sub-task coherence estimates as well as purity estimates. We observe that the proposed TE-coh-ddCRP model outperforms all compared approaches in terms of purity and is able to find subtask clusters wherein the query terms are more similar. It improves over the second best method by over 20%. We observe similar trends in terms of top performing baselines as we did while evaluating coherence. Among the variants of the proposed approach, the coherence enabled version performs better than the embedding enabled variant while their combination performs the best. This highlights the importance of considering both the embedding based distance metric aspect as well as the coherence based likelihood aspect, while extracting subtasks.

In addition to decomposing complex tasks into subtasks, we considered the

challenge of extracting hierarchies of search tasks and their associated subtasks from a given search log given just the log data without the need of any manual annotation of any sort. In Chapter 5, we presented an efficient Bayesian non-parametric model for discovering hierarchies and proposed a tree based bayesian hierarchial task construction algorithm to discover this rich hierarchical structure embedded within search logs.

We used a number of evaluation methodologies to evaluate the efficacy of the proposed task extraction methodology, including quantitative and qualitative analyses along with crowdsourced judgment studies specifically catered to evaluating the quality of the extracted task hierarchies. First, to justify the effectiveness of the proposed model in identifying search tasks in query logs, we employ a commonly used AOL data subset with search tasks annotated which is a standard test dataset for evaluating task extraction systems. The proposed approach manages to outperform existing task extraction baselines while having much greater expressive powers and providing the subdivision of tasks into subtasks. While there are no gold standard datasets for evaluating hierarchies of tasks, we performed crowd-sourced assessments to assess the performance of our hierarchy extraction method. We separately evaluated the coherence and quality of the extracted hierarchies via two different set of judgements obtained via crowdsourcing. We measured *Task Relatedness*, which measures how pure the task clusters obtained are, with a higher score indicating that the queries belonging to the same task are indeed used for solving the same search task. The overall results indicate that the tasks extracted by the proposed task-subtask extraction algorithm are indeed better than those extracted by the baselines. Additionally, we asked the human annotators to judge the subtask validity and usefulness. The identified subtask was found useful in 67% cases with the best performing baseline being useful in 52% of judged instances. This highlights that the extracted hierarchy is indeed composed of better subtasks which are found to be useful in completing the overall task depicted by the parent task.

In addition to task extraction and user study based evaluation, we chose to follow an indirect evaluation approach based on Query Term Prediction wherein

given an initial set of queries, we predict future query terms the user may issue later in the session. This is in line with our goal of supporting users tackling complex search tasks since a task identification system which is capable of identifying good search tasks will indeed perform better in predicting the set of future query terms. We demonstrated that the proposed method is able to better predict future query terms than a standard task extraction baseline as well as a very recent hierarchy construction algorithm.

Part II: Leveraging Task Information

In Part II of the thesis, we highlighted a number of ways in which the extracted task information could be used in different applications. In Chapter 6, we presented an approach to model user's in terms of their tasks by coupling user's topical interest information with their search task information to learn a joint user representation technique. We demonstrated that coupling user's task information with their topical interests indeed helps us build better user models. In order to evaluate the performance of the proposed task based user modelling techniques, we used three techniques of evaluation based on collaborative query recommendation, query recommendation based on user groups and user cohort analysis.

Our results showed that the proposed Topic-Task Tensor based user modelling approach and the coupled matrix factorization method performs better than the term based similarity method (TermSim) as well as topic based method, which demonstrates that combining search task information with user's topical interests thus help us better capture different aspects of user profiles and can serve as potent user modelling tools. Since TermSim relies strictly on term matching for measuring user similarities, its coverage is limited: it might not capture insights for the users with too few queries or those who shared the same search interest but issued different queries or performed different tasks. Task based user modelling can help in better differentiating between users which have similar topical interests but perform different tasks. The proposed tensor based approach combines the best of both the worlds and hence was able to leverage the topical user profile information with the

task aspect. Additionally, the coupled matrix-tensor method (CMTF) combines information from all available data modalities and learns a joint user representation. We see that the CMTF model outperforms the other methods which highlights the importance of jointly considering user's term, topic and task information.

Experimental results also indicated that the user clusters obtained from via using topic-task coupled representations perform better than the clusters obtained via just Bag-of-Terms or task baselines. This is in line with our hypothesis that capturing task behaviors across user populations indeed helps us in forming well-knit user clusters and thus could help us perform better in *groupization*.

A direct result of users being engaged in multitasking and task switching behaviors is that the resulting search context is heterogeneous, composed of interleaved search goals and tasks. Aimed at mitigating the ill-effects of such heterogeneous contexts, in Chapter 8 we proposed a novel task based embedding architecture to learn distributed semantic representation of query terms which prefers task context over local information in immediately surrounding words. We contributed the idea that word embeddings be learned on a task-constrained context instead of the traditionally used global or session context. We demonstrated that the proposed task embedding model is able to extract improved query representations which capture task context. In addition to qualitative analysis, we demonstrate the benefit of learning task based embeddings over traditional query representation techniques by showing enhanced performance when generating query suggestions.

We observe that though global representations perform better than traditional and simpler representation techniques, they perform worse than session based and task based embeddings. This highlights the importance of considering local context when learning representations, since generic contexts are usually heterogeneous and ill fitted to retrieval problems. Among the neural local context models, task based context performs better than session based contexts. This confirms our hypothesis that sessions are usually polluted with queries from various tasks, and as a result the resulting context isn't informative enough. While relevance scores are important, often system designers have a constraint to rank top-k suggestions. To this end,

in addition to average relevance scores, we make use of the candidate ranking to compute NDCG scores. Similar to our previous observation, we observe that neural representation methods generally perform better than non-neural models. Amidst session based and task context based, task based representation performs better than the corresponding session context.

Beyond user models, personalization and query representations, we considered using task information for predicting user satisfaction. A major portion of existing work on modelling searcher satisfaction has focused on query level satisfaction. The few existing approaches for task satisfaction prediction have narrowly focused on simple tasks aimed at solving atomic information needs. In Chapter 9, we go beyond such atomic tasks and considered the problem of predicting user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks.

We worked on a random sample of user sessions from a major US commercial search engine engine during a week in June 2016 comprising of over 14670 search sessions resulting in about 148561 search queries. We considered a holistic view of user interactions with the search engine result page (SERP) and extracted detailed interaction sequences of their activity. We then looked at query level abstraction and proposed a novel deep sequential architecture which leverages the extracted interaction sequences to predict query level satisfaction. Further, we enriched this model with auxiliary features which have been traditionally used for satisfaction prediction and proposed a unified multi-view model which combines the benefit of user interaction sequences with auxiliary features. Finally, we go beyond query level abstraction and considered query sequences issued by the user in order to complete a complex task, to make task level satisfaction predictions. We proposed a number of functional composition techniques which take into account query level satisfaction estimates along with the query sequence to predict task level satisfaction.

We conducted a number of experiments using crowdsourced judgments as well as real world search engine traffic. We made use of labels obtained via crowdsourced judgments studies as ground truth labels for all evaluations considered; however, we leveraged large scale pseudo-labeled data with weak supervision sig-

nals to train our deep models. As our first experiment, we considered predicting user satisfaction for each search impression. We compared the proposed sequential action embedding + LSTM model with traditionally used features as well as other popular feature based and sequential models. We observed that sequence based baselines perform better than feature based baselines in general, with the generative probabilistic baseline performing particularly better with over 7% improvement in accuracy scores. A satisfying click, i.e., click followed by a long dwell time, has traditionally been used to gauge user satisfaction. We re-confirm such known insights since we observe that Click+DwellTime obtain the best precision; however this method misses out on capturing various other satisfactory interactions, as is evident from their low recall scores.

We observed that the proposed deep sequential model and its variants outperform all baselines considered in predicting user satisfaction and register an improvement in over 11% over the worst performing baseline and 5% over the best performing generative sequence modelling approach. Among the variants considered, the simple RNN model is outperformed by the more sophisticated LSTM models which confirms known benefits offered by LSTMs over RNNs. The bidirectional version of the proposed model outperformed the LSTM model on all metrics, which confirmed our hypothesis that including future action signal information helps in modelling user interaction better.

Additionally, we evaluated the benefit of unifying the different interactions signals, both static features and interaction sequences. We investigated how adding different sets of features to the sequential model helps in better predicting user satisfaction. We observed that adding the other view of user interaction data always helps in improving prediction performance across all methods. Adding click based signal information to the interaction sequence information improves SAT precision (at the cost of recall), which is consistent with what was observed before. Adding temporal signals give a significantly improved performance in terms of recall, with over 27% improvement in detecting satisfaction cases which may have otherwise been missed by baseline approaches. Further, the unified multi-view model achieves

the best accuracy in predicting user satisfaction with over 5% improvement in accuracy, 26% improvement in recall and 7% improvement in F-score.

One major motivation for the work presented in Chapter 9 was to leverage user interaction signals to predict task level satisfaction of users. To this end, we considered the problem of task satisfaction prediction and compared how the different compositional functions perform in predicting task level satisfaction. We observed that the proposed deep sequential model performs better than the best performing baselines in predicting task satisfaction across all five metrics considered. Moreover, the unified multi-view model performs better than the deep sequential model, which demonstrates that the combined information from interaction sequences and other auxiliary implicit feedback signals are not only good for query level satisfaction prediction, but also work best at measuring task satisfaction. Comparison of the different functional compositional techniques for task SAT prediction highlighted that the most lenient aggregating technique (Maximum) consistently achieves higher accuracy than the most strict satisfaction criterion (Minimum). We also observed that the differential weighting scheme performs better than the average function, which hints at the fact that not all queries contribute the same towards a task. Finally, considering subtasks information in the intermediary stage between query and task level abstractions helps in better predicting task satisfaction.

9.2 Implications

In this section, we discuss some of the implications of the research presented in the different chapters of this thesis. The findings on user's multitasking behavior presented in Chapter 3 are useful for search engines in that they could reduce task-transition delays and make design improvements to reduce cognitive loads in multi-task sessions. Insights about the presence of various user groups based on searcher's multi-tasking habits could inspire the development of personalized models for different user groups based on user's inherent multi-tasking habits. Additionally, the insights presented about user, interest and topic level heterogeneities in search behavior motivates the need for a shift in focus from search sessions to search tasks as

the primary focal unit of consideration and analysis.

The task-subtask extraction methods presented in Chapters 4 and 5 have implications not just in the area of web search but also in other digital online user facing services. Understanding the task a user is trying to accomplish would assist the search engine to make better task-aware query suggestions, help develop improved personalization models, provide better recommendations not just for the current subtask but also for any future subtask the user might attempt. Such proactive task aware recommendations could propel a paradigm shift in web search from a reactive interaction towards more proactive interactions. Moreover, accurate representations of tasks could also be highly useful in aptly placing the user in the task-subtask space to contextually target the user in terms of better recommendations and advertisements, developing task specific ranking of documents, and developing task based evaluation metrics to model user satisfaction.

In addition to offering improved services, task awareness could inspire research and development of novel task aware interfaces which shift the focus of search engines from single query information solving tools to task completion engines.

The research presented in Chapter 6 on task based personalization encourages researchers to appreciate the benefits of task information while building user models, and motivates the need for developing task aware user representations. User modelling is common to all online services which aim to offer personalized services. Given the fact that users use these services to perform certain tasks, developing task based representations would enable service provides to personalize and assist their users in successfully using their services. The successful results on user cohorts and user modelling presented in Chapter 6 clearly demonstrates the value of considering search tasks rather than just query terms or topics during personalization.

The research presented in Chapter 7 considers task context for learning query representations. The findings presented in this chapter imply that tasks context enriched representations perform better than traditional representations, and at the same time, task context is more informative than session context. Since search

context is heavily used in a number of search services including user interest modelling, search re-ranking, query suggestions etc, these findings have implications in designing better context-sensitive search models aimed at exploiting task context for enhanced user support.

Beyond user modelling and query representations, this thesis considers the problem of predicting task based user satisfaction in Chapter 9. The implication of our finding is that detailed view of user interactions helps in predicting not just query level but also task level user satisfaction. Going beyond atomic tasks and considering the problem of user's satisfaction when engaged in complex search tasks composed of many different queries and subtasks, enables system designers to develop metrics which capture user's overall satisfaction in completing the task which inspired them to engage with the search system in the first place. Furthermore, the findings of this work could inspire future research in developing sophisticated techniques for quantifying the importance of different queries and subtasks in any given complex task. Further, task satisfaction prediction could inspire research in developing retrieval algorithms optimized for task completion. Finally, we contend that the promising results demonstrated by the unified multi-view approach would help in improving satisfaction prediction and good abandonment detection on mobile devices.

Finally, it is important to note that the notion of tasks is not limited to web search space, but easily transcends the search domain and has wide applicability in other user centric online services. Task aware models and metrics could help system designers in specifically targeting users based on the tasks they are involved in. We believe that insights from this work could spark future research in developing richer and generalizable models of tasks beyond web search.

9.3 Limitations

While the methods and results presented in the thesis advance our understanding of search tasks and advance the state-of-the-art in task extraction, the work presented has certain limitations which warrant further investigation in future research. One

major limitation of the extracted tasks is *interpretability*. The extracted tasks often comprise of a collection of queries without any textual description of what the task is about. This limits the applicability of the tasks to user facing systems wherein we want to show task summaries to the user. Another limitation of the proposed algorithms is exhibited the lack of task sequence data, which prohibits us from investigating and proposing task sequence prediction algorithms. Indeed, both at the task level and the subtask level, often times there exists an underlying sequence which is preferred by users. Identifying and leveraging such sequential information would help us better plan user's journey toward task completion.

A related aspect about subtasks is information about their attributes. Certain subtasks could be optional, or compulsory; real world task or digital tasks; easy or difficult to complete. The algorithms presented in this thesis do not leverage such attributes of subtasks and are hence limited in exploiting these aspects to better assist users. Finally, this thesis considers and covers web search as the focal domain for grounding research on task understanding and extraction. While information about search tasks is indeed very useful in web search context, the notion of tasks extend naturally to most other digital activity of the user across different devices and platforms. This thesis doesn't not presents results on task based recommendations or task based interfaces. Finally, as increasingly more users rely on mobile devices to complete their tasks, it becomes important to incorporate mobile specific nuances in the task extraction models. The research presented in this thesis mainly focuses on query text and in some instances, on the presence of mouse movement features, and thus, incorporation of mobile specific features would need to be addressed in future work. Given these limitations, we next discuss few potential areas of future research.

9.4 Future Work

The research presented in this thesis has many important implications, as described above. However, there are a number of opportunities for further work. In this section we select the most prominent directions and summarize them.

9.4.1 Task based Conversational Intelligence

While reactive search by issuing query remains as common as before, recent years have witnessed the emergence of various proactive systems in the form of digital assistants such as Siri, Google Now and Microsoft Cortana. Such systems make use of a plethora of signals including user's input query and contextual information to provide assistance by answering questions in natural language, making recommendations, and performing actions. An important outcome of the research presented in this thesis is task understanding and extraction. Parts of the algorithms presented in Part I of this thesis could be adapted to understand user's task from conversational context. Given the task information, an intelligent task aware conversational module could be build by enriching current conversational context with domain specific task information, thereby constructing intelligent verbal responses from user context, task domain knowledge and current information needs of the user. Imparting task awareness to any chatbot is a completely open and promising area not just for research but also for consumer centric industrial applications.

9.4.2 Extracting Sub-Task Sequences

While web search researchers have investigated benefits of incorporating sequence predictions for enhanced search support, huge gaps still remain. Recent efforts have highlighted the importance of considering search trails for enhancing search support. Recent studies have quantified the benefit that users currently obtain from trail following and compare different methods for finding the best trail for a given query and each top-ranked result. An important step in developing task based systems is automatically identifying such sequences and aspects of complex tasks which searchers engage in. While the research presented in Part I of this thesis helps in extracting the subtasks forming the complex task, an important problem in developing a task aware system is to find out the order (or sequence) of the subtasks. Specifically, this research direction entails developing sequence extraction and prediction algorithms capable of scanning web search logs and identifying the different sub-task sequences which constitute various aspects and steps of the different search tasks. Identifying sub-task (and task) sequences would tremendously bene-

fit systems such as task tours since by making use of the extracted sequences, the quality of the task tours developed could be improved and the tours themselves be personalized.

9.4.3 Metrics for Evaluating Hierarchies

Evaluating the quality of a task hierarchy is a non-trivial task, the ultimate test being user judgment. While there is no concrete standardized metric used in evaluating hierarchies, existing clustering based literature has made use of FScore to evaluate the accuracy with which the documents are assigned to the clusters. While evaluating task hierarchy extraction systems, there are two scenarios to consider: (i) with tagged query-task gold standard dataset and (ii) unsupervised evaluation. Expected outcomes include a set of evaluation metrics for both the cases where gold standard dataset is available and when its not. A good future research direction would evaluate all the existing task extraction systems on the proposed metrics and additionally present a survey of the task understanding field and existing task extraction algorithms based on the detailed experimentation on such new task-aware metrics.

9.4.4 Modelling Tasks beyond Web Search

The concept of tasks in generalizable across different fields and systems built around user interactions. An important future direction to consider would be to model general user tasks across devices, applications and systems to better target users with enhanced forms of contextual personalization. As stated earlier, the notion of tasks extends well beyond web search, with each user action with an online service hinting towards certain task the user is trying to accomplish. Identifying such generic task spaces in different application domains and developing generic task extraction systems could benefit many different online services in better serving their users.

Beyond the above mentioned applications, task information can be leveraged to develop novel task completion interfaces, develop guides for users to support and complete their tasks, among others.

Appendix A

Full List of Publications

Rishabh Mehrotra has published the following conference papers during his Ph.D.

1. Rishabh Mehrotra, Emine Yilmaz. *Task Embeddings: Learning Query Embeddings using Task Context*. In proceedings of CIKM 2017.

2. Rishabh Mehrotra, Ahmed Hassan Awadallah, Milad Shokouhi, Emine Yilmaz, Imed Zitouni, Ahmed El Kholy and Madian Khabsa. *Deep Sequential Models for Task Satisfaction Prediction*. In proceedings of CIKM 2017.

3. Rishabh Mehrotra and Prasanta Bhattacharya. *Characterizing and Predicting Supply-side Engagement on Crowd-contributed Video Sharing Platforms*. In proceedings of ICTIR 2017.

4. Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan Awadallah and Ahmed El Kholy. *Extracting Leveraging User Interaction Sequences for Search Satisfaction Prediction* In proceedings of SIGIR 2017.

5. Rishabh Mehrotra and Emine Yilmaz. *Extracting Hierarchies of Search Tasks Subtasks via a Bayesian Nonparametric Approach*. In proceedings of SIGIR 2017.

6. Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach and Emine Yilmaz. *Auditing Search Engines for Differential Performance Across Demographics*. In proceedings of WWW 2017.

7. Rishabh Mehrotra, Prasanta Bhattacharya, Tianhui Tan and Tuan Q. Phan. *Predictive Power of Online and Offline Behavior Sequences: Evidence from a Microfinance Context*. In proceedings of ICIS 2017.

8. Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan Awadallah, Milad Shokouhi and Ahmed El Kholy. *Identifying User Sessions in Interactions with Intelligent Assistants*. In proceedings of WWW 2017 (Posters Track).

9. Rishabh Mehrotra and Emine Yilmaz. *Query Log Mining for Inferring User Tasks and Needs*. In proceedings of ECML 2016 (Nectar Track).

10. Rishabh Mehrotra, Prasanta Bhattacharya and Emine Yilmaz. *Uncovering Task Based Behavioral Heterogeneities in Online Search Behavior*. In proceedings of SIGIR 2016.

11. Rishabh Mehrotra, Prasanta Bhattacharya and Emine Yilmaz. *Deconstructing Complex Search Tasks*. In proceedings of NAACL 2016.

12. Rishabh Mehrotra, Prasanta Bhattacharya and Emine Yilmaz. *Characterizing Users' Multi-Tasking Behavior in Web Search*. In proceedings of CHIIR 2016.

13. Prasanta Bhattacharya, Rishabh Mehrotra. *The Information Network: Exploiting Causal Dependencies in Online Information Seeking*. In proceedings of CHIIR 2016.

14. Rishabh Mehrotra and Emine Yilmaz. *Representative Informative Query Selection for Learning to Rank using Submodular Functionsd Needs*. In proceedings of SIGIR 2015.

15. Rishabh Mehrotra and Emine Yilmaz. *Terms, Topics Tasks: Enhanced User Modelling for Better Personalization*. In proceedings of ICTIR 2015.

16. Rishabh Mehrotra, Prasanta Bhattacharya. *Modeling the Evolution of User-generated Content on a Large Video Sharing Platform*. In proceedings of WWW 2015 (Web Science Track).

17. Rishabh Mehrotra and Emine Yilmaz. *Towards Hierarchies of Search Tasks Subtasks*. In proceedings of WWW 2015 (Posters track).

18. Emine Yilmaz, Evangelos Kanoulas, Manisha Verma, Nick Craswell, Rishabh Mehrotra. *Overview of the TREC 2015 Tasks Track*. In proceedings of TREC 2015.

19. Rishabh Mehrotra. *Topics, Tasks Beyond: Learning Representations for Personalization*. In proceedings of WSDM 2015 (Doctoral Consortium).

20. Rishabh Mehrotra and Emine Yilmaz. *Task-Based User Modelling for Personalization via Probabilistic Matrix Factorization*. In proceedings of RecSys 2014.

Bibliography

[1] Search engine statistics 2017. http://www.smartinsights.com/search-engine-marketing/search-engine-statistics/. Accessed: 2017-09-12.

[2] How google search works. https://www.google.com/search/howsearchworks/. Accessed: 2017-09-12.

[3] Herman Tavani. Search engines and ethics. 2012.

[4] Nicholas J Belkin. A methodology for taking account of user tasks, goals and behavior for design of computerized library catalogs. *ACM SIGCHI Bulletin*, 23(1):61–65, 1991.

[5] Debora Donato, Francesco Bonchi, Tom Chi, and Yoelle Maarek. Do you want to take notes?: identifying research missions in yahoo! search pad. In *Proceedings of the 19th international conference on World wide web*, pages 321–330. ACM, 2010.

[6] Alexander Kotov, Paul N Bennett, Ryen W White, Susan T Dumais, and Jaime Teevan. Modeling and analysis of cross-session search tasks. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 5–14. ACM, 2011.

[7] Jingjing Liu and Nicholas J Belkin. Personalizing information retrieval for multi-session tasks: The roles of task stage and task type. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 26–33. ACM, 2010.

[8] Ryen W White, Paul N Bennett, and Susan T Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1009–1018. ACM, 2010.

[9] Ahmed Hassan Awadallah, Ryen W White, Patrick Pantel, Susan T Dumais, and Yi-Min Wang. Supporting complex search tasks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 829–838. ACM, 2014.

[10] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W White, and Wei Chu. Learning to extract cross-session search tasks. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1353–1364. ACM, 2013.

[11] Yuelin Li. *Relationships among work tasks, search tasks, and interactive information searching behavior*. Rutgers The State University of New Jersey-New Brunswick, 2008.

[12] Pertti Vakkari. Task-based information searching. *Annual review of information science and technology*, 37(1):413–464, 2003.

[13] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 699–708. ACM, 2008.

[14] Emine Yilmaz, Manisha Verma, Rishabh Mehrotra, Evangelos Kanoulas, Ben Carterette, and Nick Craswell. Overview of the trec 2015 tasks track. In *TREC*, 2015.

[15] Manisha Verma, Emine Yilmaz, Rishabh Mehrotra, Evangelos Kanoulas, Ben Carterette, Nick Craswell, and Peter Bailey. Overview of the trec tasks track 2016. In *TREC*, 2016.

[16] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. Characterizing users' multi-tasking behavior in web search. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, pages 297–300. ACM, 2016.

[17] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. Uncovering task based behavioral heterogeneities in online search behavior. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1049–1052. ACM, 2016.

[18] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. Deconstructing complex search tasks: a bayesian nonparametric approach for extracting sub-tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 599–605, 2016.

[19] Rishabh Mehrotra and Emine Yilmaz. Extracting hierarchies of search tasks & subtasks via a bayesian nonparametric approach. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 285–294. ACM, 2017.

[20] Rishabh Mehrotra and Emine Yilmaz. Towards hierarchies of search tasks & subtasks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 73–74. ACM, 2015.

[21] Rishabh Mehrotra and Emine Yilmaz. Terms, topics & tasks: Enhanced user modelling for better personalization. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 131–140. ACM, 2015.

[22] Rishabh Mehrotra and Emine Yilmaz. Task embeddings: Learning query embeddings using task context. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2199–2202. ACM, 2017.

[23] Rishabh Mehrotra, Ahmed Hassan Awadallah, Milad Shokouhi, Emine Yilmaz, Imed Zitouni, Ahmed El Kholy, and Madian Khabsa. Deep sequential models for task satisfaction prediction. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 737–746. ACM, 2017.

[24] Rishabh Mehrotra, Imed Zitouni, Ahmed Hassan Awadallah, Ahmed El Kholy, and Madian Khabsa. User interaction sequences for search satisfaction prediction. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174. ACM, 2017.

[25] Rishabh Mehrotra, A Hassan Awadallah, AE Kholy, and Imed Zitouni. Hey cortana! exploring the use cases of a desktop based digital assistant. In *SIGIR 1st International Workshop on Conversational Approaches to Information Retrieval (CAIR'17)*, volume 4, 2017.

[26] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. Auditing search engines for differential satisfaction across demographics. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 626–633. International World Wide Web Conferences Steering Committee, 2017.

[27] Rishabh Mehrotra and Prasanta Bhattacharya. Characterizing and predicting supply-side engagement on video sharing platforms using a hawkes process model. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 159–166. ACM, 2017.

[28] Rishabh Mehrotra and Emine Yilmaz. Representative & informative query selection for learning to rank using submodular functions. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 545–554. ACM, 2015.

[29] Prasanta Bhattacharya and Rishabh Mehrotra. The information network: Exploiting causal dependencies in online information seeking. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, pages 223–232. ACM, 2016.

[30] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892. ACM, 2013.

[31] Rishabh Mehrotra, Rushabh Agrawal, and Syed Aqueel Haider. Dictionary based sparse representation for domain adaptation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2395–2398. ACM, 2012.

[32] Syed Aqueel Haider and Rishabh Mehrotra. Corporate news classification and valence prediction: A supervised approach. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 175–181. Association for Computational Linguistics, 2011.

[33] Simon Eliot and Jonathan Rose. *A Companion to the History of the Book*, volume 98. John Wiley & Sons, 2009.

[34] Soper Herbert Edward. Means for compiling tabular and statistical data, August 31 1920. US Patent 1,351,692.

[35] Vannevar Bush et al. As we may think. *The atlantic monthly*, 176(1):101–108, 1945.

[36] Calvin N Mooers. *The theory of digital handling of non-numerical information and its implications to machine economics*. Number 48. Zator Co., 1950.

[37] JE Holmstrom. Section iii. opening plenary session. In *The Royal Society Scientific Information Conference, 21 June–2 July 1948: Report and papers submitted*, 1948.

[38] Amit Singhal et al. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[39] D Harmon. Overview of the first text retrieval conference (trec-1). *NIST Special Publication*, pages 500–207.

[40] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[41] Gerard Salton. Automatic information organization and retrieval. 1968.

[42] Cyril W Gleverdon and Cyril W Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. 1962.

[43] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press, 2008.

[44] Nicholas J Belkin, Robert N Oddy, and Helen M Brooks. Ask for information retrieval: Part i. background and theory. *Journal of documentation*, 38(2):61–71, 1982.

[45] Peter Ingwersen and Kalervo Järvelin. *The turn: Integration of information seeking and retrieval in context*, volume 18. Springer Science & Business Media, 2006.

[46] Yuelin Li and Nicholas J Belkin. A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management*, 44(6):1822–1837, 2008.

[47] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building bridges for web query classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 131–138. ACM, 2006.

[48] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *Informs journal on computing*, 15(2):171–190, 2003.

[49] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and information systems*, 1(1):5–32, 1999.

[50] Lara D Catledge and James E Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN systems*, 27(6):1065–1073, 1995.

[51] Carsten Eickhoff, Jaime Teevan, Ryen White, and Susan Dumais. Lessons from the journey: a query log analysis of within-session learning. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 223–232. ACM, 2014.

[52] José Luis Ortega and Isidro Aguillo. Differences between web sessions according to the origin of their visits. *Journal of Informetrics*, 4(3):331–337, 2010.

[53] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248. ACM, 2005.

[54] Bhaskar Mitra. Exploring session context using distributed representations of queries and reformulations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 3–12. ACM, 2015.

[55] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2005.

[56] Diane Kelly, Vijay Deepak Dollu, and Xin Fu. The loquacious user: a document-independent source of terms for query expansion. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 457–464. ACM, 2005.

[57] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883. ACM, 2008.

[58] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. Context-aware query classification. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 2009.

[59] Lilyana Mihalkova and Raymond Mooney. Learning to disambiguate search queries from short sessions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 111–127. Springer, 2009.

[60] Luanne Freund. *Exploiting task-document relations in support of information retrieval in the workplace*. University of Toronto, 2008.

[61] Melanie Kellar, Carolyn Watters, and Michael Shepherd. A field study characterizing web-based information-seeking tasks. *Journal of the Association for Information Science and Technology*, 58(7):999–1018, 2007.

[62] Reijo Savolainen. Everyday life information seeking: Approaching information seeking in the context of way of life. *Library & information science research*, 17(3):259–294, 1995.

[63] SJ Lin. Modeling and supporting multiple information seeking episodes over the web. *Unpublished dissertation. Rutgers University*, 2001.

[64] Denise E Agosto and Sandra Hughes-Hassell. People, places, and questions: An investigation of the everyday life information-seeking behaviors of urban young adults. *Library & information science research*, 27(2):141–163, 2005.

[65] Nicholas J Belkin, Michael Cole, and Jingjing Liu. A model for evaluation of interactive information retrieval. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 7–8, 2009.

[66] Gary Marchionini. Information-seeking strategies of novices using a full-text electronic encyclopedia. *Journal of the American Society for Information Science*, 40(1):54, 1989.

[67] Liwen Qiu. Analytical searching vs. browsing in hypertext information retrieval systems. *Canadian Journal of Information and Library Science*, 18(4):1–13, 1993.

[68] Jeonghyun Kim. *Task as a predictable indicator for information seeking behavior on the Web*. ProQuest, 2006.

[69] Elaine Toms, Tayze MacKenzie, Chris Jordan, H OBrien, L Freund, Sandra Toze, Emille Dawe, and Alexandra MacNutt. How task affects information search. In *Workshop Pre-proceedings in Initiative for the Evaluation of XML Retrieval (INEX)*, pages 337–341, 2007.

[70] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Identifying task-based sessions in search engine query logs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 277–286. ACM, 2011.

[71] Amanda Spink, Sherry Koshman, Minsoo Park, Chris Field, and Bernard J Jansen. Multitasking web search on vivisimo. com. In *Information Technol-*

ogy: Coding and Computing, 2005. ITCC 2005. International Conference on, volume 2, pages 486–490. IEEE, 2005.

[72] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Detecting task-based query sessions using collaborative knowledge. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 3, pages 128–131. IEEE, 2010.

[73] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Discovering tasks from search engine query logs. *ACM Transactions on Information Systems (TOIS)*, 31(3):14, 2013.

[74] Eugene Agichtein, Ryen W White, Susan T Dumais, and Paul N Bennet. Search, interrupted: understanding and predicting search task continuation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 315–324. ACM, 2012.

[75] Zhen Liao, Yang Song, Li-wei He, and Yalou Huang. Evaluating the effectiveness of search task trails. In *Proceedings of the 21st international conference on World Wide Web*, pages 489–498. ACM, 2012.

[76] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, and Hongyuan Zha. Identifying and labeling search tasks via query-based hawkes processes. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–740. ACM, 2014.

[77] Gabriele Tolomei, Salvatore Orlando, and Fabrizio Silvestri. Towards a task-based search and recommender systems. In *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, pages 333–336. IEEE, 2010.

[78] Janette Lehmann, Mounia Lalmas, Georges Dupret, and Ricardo Baeza-Yates. Online multitasking and user engagement. In *Proceedings of the 22nd*

ACM international conference on Information & Knowledge Management, pages 519–528. ACM, 2013.

[79] Qing Wang and Huiyou Chang. Multitasking bar: prototype and evaluation of introducing the task concept into a browser. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 103–112. ACM, 2010.

[80] Amanda Spink, Minsoo Park, Bernard J Jansen, and Jan Pedersen. Multitasking during web search sessions. *Information Processing & Management*, 42(1):264–275, 2006.

[81] Ravi Kumar and Andrew Tomkins. A characterization of online browsing behavior. In *Proceedings of the 19th international conference on World wide web*, pages 561–570. ACM, 2010.

[82] Hartmut Obendorf, Harald Weinreich, Eelco Herder, and Matthias Mayer. Web page revisitation revisited: implications of a long-term click-stream study of browser usage. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 597–606. ACM, 2007.

[83] Bonnie Ma Kay and Carolyn Watters. Exploring multi-session web tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1187–1196. ACM, 2008.

[84] Dan Morris, Meredith Ringel Morris, and Gina Venolia. Searchbar: a search-centric web history for task resumption and information re-finding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1207–1216. ACM, 2008.

[85] Brendan O'Connor, Michel Krieger, and David Ahn. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*, pages 384–385, 2010.

[86] Adish Singla, Ryen White, and Jeff Huang. Studying trailfinding algorithms for enhanced web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 443–450. ACM, 2010.

[87] Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th international conference on World wide web*, pages 191–200. ACM, 2009.

[88] Ranieri Baraglia, Fidel Cacheda, Victor Carneiro, Vreixo Formoso, Raffaele Perego, and Fabrizio Silvestri. Search shortcuts: Driving users towards their goals. In *Proceedings of the 18th international conference on World wide web*, pages 1073–1074. ACM, 2009.

[89] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Modeling and predicting the task-by-task behavior of search engine users. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 77–84. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013.

[90] Nicolaas Matthijs and Filip Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 25–34. ACM, 2011.

[91] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590. ACM, 2007.

[92] Morgan Harvey, Fabio Crestani, and Mark J Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2309–2314. ACM, 2013.

[93] Paul N Bennett, Ryen W White, Wei Chu, Susan T Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the impact of short-and long-term behavior on search personalization. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 185–194. ACM, 2012.

[94] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Beyond the commons: Investigating the value of personalizing web search. In *Proceedings of the Workshop on New Technologies for Personalized Information Access (PIA)*, pages 84–92, 2005.

[95] Hongning Wang, ChengXiang Zhai, Feng Liang, Anlei Dong, and Yi Chang. User modeling in search logs via a nonparametric bayesian approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 203–212. ACM, 2014.

[96] Smitha Sriram, Xuehua Shen, and Chengxiang Zhai. A session-based search engine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 492–493. ACM, 2004.

[97] Mirco Speretta and Susan Gauch. Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. IEEE, 2005.

[98] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web*, pages 727–736. ACM, 2006.

[99] Qi Guo, Ryen W White, Susan T Dumais, Jue Wang, and Blake Anderson. Predicting query performance using query, result, and user interaction features. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*, pages 198–201. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2010.

[100] Qi Guo, Dmitry Lagun, and Eugene Agichtein. Predicting web search success with fine-grained interaction data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2050–2054. ACM, 2012.

[101] Qi Guo and Eugene Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proceedings of the 21st international conference on World Wide Web*, pages 569–578. ACM, 2012.

[102] Jeff Huang, Ryen W White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 195–204. ACM, 2012.

[103] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI'08 extended abstracts on Human factors in computing systems*, pages 2997–3002. ACM, 2008.

[104] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 707–708. ACM, 2008.

[105] Yiqun Liu, Chao Wang, Ke Zhou, Jianyun Nie, Min Zhang, and Shaoping Ma. From skimming to reading: A two-stage examination model for web search. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 849–858. ACM, 2014.

[106] Jeff Huang, Ryen W White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1225–1234. ACM, 2011.

[107] Ioannis Arapakis, Mounia Lalmas, and George Valkanas. Understanding within-content engagement through pattern analysis of mouse gestures. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1439–1448. ACM, 2014.

[108] Dmitry Lagun, Mikhail Ageev, Qi Guo, and Eugene Agichtein. Discovering common motifs in cursor movement data for improving web search. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 183–192. ACM, 2014.

[109] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. Beyond dcg: user behavior as a predictor of a successful search. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 221–230. ACM, 2010.

[110] Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W White. Understanding and predicting graded search satisfaction. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 57–66. ACM, 2015.

[111] Ryen W White and Susan T Dumais. Characterizing and predicting search engine switching behavior. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 87–96. ACM, 2009.

[112] Denis Savenkov, Dmitry Lagun, and Qiaoling Liu. Search engine switching detection based on user personal preferences and behavior patterns. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 33–42. ACM, 2013.

[113] Ahmed Hassan and Ryen W White. Task tours: helping users tackle complex search tasks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1885–1889. ACM, 2012.

[114] Yiqun Liu, Ye Chen, Jinhui Tang, Jiashen Sun, Min Zhang, Shaoping Ma, and Xuan Zhu. Different users, different opinions: Predicting search sat-

isfaction with mouse movement information. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 493–502. ACM, 2015.

[115] Louise T Su. Evaluation measures for interactive information retrieval. *Information Processing & Management*, 28(4):503–516, 1992.

[116] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(12):1–224, 2009.

[117] Henry A Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41. ACM, 2010.

[118] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

[119] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2019–2028. ACM, 2013.

[120] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 895–898. ACM, 2014.

[121] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 113–122. ACM, 2014.

[122] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[123] 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*, pages 575–584. ACM, 2015.

[124] Ivan Vulić and Marie-Francine Moens. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 363–372. ACM, 2015.

[125] Parth Gupta, Kalika Bali, Rafael E Banchs, Monojit Choudhury, and Paolo Rosso. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 677–686. ACM, 2014.

[126] Tom Kenter and Maarten De Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1411–1420. ACM, 2015.

[127] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM, 2014.

[128] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. Context-and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 383–392. ACM, 2015.

[129] Bhaskar Mitra and Nick Craswell. Query auto-completion for rare prefixes. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1755–1758. ACM, 2015.

[130] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems*, pages 17–24, 2004.

[131] Jim Pitman et al. Combinatorial stochastic processes. 2002.

[132] Chien-Liang Liu, Tsung-Hsun Tsai, and Chia-Hoang Lee. Online chinese restaurant process. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 591–600. ACM, 2014.

[133] Richard Socher, Andrew Maas, and Christopher Manning. Spectral chinese restaurant processes: Nonparametric clustering based on similarities. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 698–706, 2011.

[134] Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya. Temporally coherent crp: a bayesian non-parametric approach for clustering tracklets with applications to person discovery in videos. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 801–809. SIAM, 2015.

[135] Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. Pos induction with distributional and morphological information using a distance-dependent chinese restaurant process. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 265–271, 2014.

[136] Himabindu Lakkaraju, Indrajit Bhattacharya, and Chiranjib Bhattacharyya. Dynamic multi-relational chinese restaurant process for analyzing influences on users in social media. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 389–398. IEEE, 2012.

[137] Charles Blundell and Yee Whye Teh. Bayesian hierarchical community discovery. In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2013.

[138] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1433–1441. ACM, 2012.

[139] Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.

[140] Shui-Lung Chuang and Lee-Feng Chien. Towards automatic generation of query taxonomy: A hierarchical query clustering approach. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 75–82. IEEE, 2002.

[141] Hui Yang. Constructing task-specific taxonomies for document collection browsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1278–1289. Association for Computational Linguistics, 2012.

[142] Hui Yang. Browsing hierarchy construction by minimum evolution. *ACM Transactions on Information Systems (TOIS)*, 33(3):13, 2015.

[143] Dawn J Lawrie and W Bruce Croft. Generating hierarchical summaries for web searches. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 457–458. ACM, 2003.

[144] Eran Segal, Daphne Koller, and Dirk Ormoneit. Probabilistic abstraction hierarchies. In *Advances in Neural Information Processing Systems*, pages 913–920, 2002.

[145] Charles Blundell, Yee Whye Teh, and Katherine A Heller. Bayesian rose trees. *arXiv preprint arXiv:1203.3468*, 2012.

[146] Ledyard R Tucker. The extension of factor analysis to three-dimensional matrices. *Contributions to mathematical psychology*, 110119, 1964.

[147] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[148] Alwin Stegeman and Nicholas D Sidiropoulos. On kruskals uniqueness condition for the candecomp/parafac decomposition. *Linear Algebra and its applications*, 420(2-3):540–552, 2007.

[149] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[150] Lieven De Lathauwer. A survey of tensor methods. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2773–2776. IEEE, 2009.

[151] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[152] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 451–458. ACM, 2010.

[153] Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem, and Bilal Chebaro. A session based personalized search using an ontological user profile. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1732–1736. ACM, 2009.

[154] Daan Odijk, Ryen W White, Ahmed Hassan Awadallah, and Susan T Dumais. Struggling and success in web search. In *Proceedings of the 24th ACM*

International on Conference on Information and Knowledge Management, pages 1551–1560. ACM, 2015.

[155] Carsten Eickhoff, Kevyn Collins-Thompson, Paul N Bennett, and Susan Dumais. Personalizing atypical web search sessions. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 285–294. ACM, 2013.

[156] Daqing He, Ayşe Göker, and David J Harper. Combining evidence for automatic web session identification. *Information Processing & Management*, 38(5):727–742, 2002.

[157] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM, 1999.

[158] Marcel Adam Just, Patricia A Carpenter, Timothy A Keller, Lisa Emery, Holly Zajac, and Keith R Thulborn. Interdependence of nonoverlapping cortical systems in dual cognitive tasks. *Neuroimage*, 14(2):417–426, 2001.

[159] Makiko Miwa. User situations and multiple levels of user goals in information problem solving processes of askeric users. In *Proceedings of the ASIST Annual Meeting*, volume 38, pages 355–71. ERIC, 2001.

[160] David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488, 2011.

[161] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 215–224. ACM, 2010.

[162] Pavel Pecina. Lexical association measures and collocation extraction. *Language resources and evaluation*, 44(1-2):137–158, 2010.

[163] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale*, volume 152, page 1, 2006.

[164] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[165] Chong Wang and David M Blei. Variational inference for the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2009.

[166] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. Association for Computational Linguistics, 2010.

[167] Ahmed Hassan Awadallah, Ryen W White, Patrick Pantel, Susan T Dumais, and Yi-Min Wang. Supporting complex search tasks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 829–838. ACM, 2014.

[168] Ryen W White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1411–1420. ACM, 2013.

[169] Yongfeng Zhang, Min Zhang, Yiqun Liu, Chua Tat-Seng, Yi Zhang, and Shaoping Ma. Task-based recommendation on a web-scale. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 827–836. IEEE, 2015.

[170] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen W White. Modeling action-level satisfaction for search task

satisfaction prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 123–132. ACM, 2014.

[171] Wen Hua, Yangqiu Song, Haixun Wang, and Xiaofang Zhou. Identifying users' topical tasks in web search. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 93–102. ACM, 2013.

[172] Qiaozhu Mei, Hui Fang, and ChengXiang Zhai. A study of poisson query generation model for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 319–326. ACM, 2007.

[173] Ben Carterette, Evangelos Kanoulas, Mark Hall, and Paul Clough. Overview of the trec 2014 session track. Technical report, DELAWARE UNIV NEWARK DEPT OF COMPUTER AND INFORMATION SCIENCES, 2014.

[174] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 699–708. ACM, 2008.

[175] Rishabh Mehrotra, Emine Yilmaz, and Manisha Verma. Task-based user modelling for personalization via probabilistic matrix factorization. In *RecSys Posters*, 2014.

[176] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422*, 2011.

[177] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[178] Jorge Nocedal and Stephen J Wright. Least-squares problems. *Numerical optimization*, pages 245–269, 2006.

[179] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of Chemometrics*, 25(2):67–86, 2011.

[180] Ehsan Abbasnejad, Scott Sanner, Edwin V Bonilla, Pascal Poupart, et al. Learning community-based preferences via dirichlet process mixtures of gaussian processes. In *IJCAI*, pages 1213–1219, 2013.

[181] Andrew Postlewaite. Social norms and social assets. *Annu. Rev. Econ.*, 3(1):239–259, 2011.

[182] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[183] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[184] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[185] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[186] Rishabh Mehrotra, Prasanta Bhattacharya, and Emine Yilmaz. Deconstructing complex search tasks: a bayesian nonparametric approach for extracting sub-tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 599–605, 2016.

[187] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*, 2016.

[188] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 193–202. ACM, 2014.

[189] Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.

[190] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[191] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[192] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[193] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. The query-flow graph: model and applications. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 609–618. ACM, 2008.

[194] Ryen W White, Matthew Richardson, and Wen-tau Yih. Questions vs. queries in informational search tasks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 135–136. ACM, 2015.

[195] Kyle Williams, Julia Kiseleva, Aidan C Crook, Imed Zitouni, Ahmed Hassan Awadallah, and Madian Khabsa. Detecting good abandonment in mobile

search. In *Proceedings of the 25th International Conference on World Wide Web*, pages 495–505. International World Wide Web Conferences Steering Committee, 2016.

[196] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[197] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[198] Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C Crook, Imed Zitouni, and Tasos Anastasakos. Predicting user satisfaction with intelligent assistants. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 45–54. ACM, 2016.

[199] Ahmed Hassan. A semi-supervised approach to modeling web search satisfaction. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 275–284. ACM, 2012.