UNIVERSITY OF CAMBRIDGE

A user-centred approach to information retrieval

Saad Muhammad S. Aloteibi

Wolfson college

This dissertation is submitted on December, 2020 for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Saad Muhammad S. Aloteibi
December, 2020

Abstract

A user-centred approach to information retrieval

Saad Muhammad S. Aloteibi

A user model is a fundamental component in user-centred information retrieval systems. It enables personalisation of a user's search experience. The development of such a model involves three phases: collecting information about each user, representing such information, and integrating the model into a retrieval application. Progress in this area is typically met with privacy and scalability challenges that hinder the ability to synthesise collective knowledge from each user's search behaviour. In this thesis, I propose a framework that addresses each of these three phases. The proposed framework is based on social role theory from the social science literature and at the centre of this theory is the concept of a social position. A social position is a label for a group of users with similar behavioural patterns. Examples of such positions are *traveller, patient, movie fan,* and *computer scientist.* In this thesis, a social position acts as a label for users who are expected to have similar interests. The proposed framework does not require real users' data; rather it uses the web as a resource to model users.

The proposed framework offers a data-driven and modular design for each of the three phases of building a user model. First, I present an approach to identify social positions from natural language sentences. I formulate this task as a binary classification task and develop a method to enumerate candidate social positions. The proposed classifier achieves an accuracy score of 85.8%, which indicates that social positions can be identified with good accuracy. Through an inter-annotator agreement study, I further show a reasonable level of agreement between users when identifying social positions.

Second, I introduce a novel topic modelling-based approach to represent each social position as a multinomial distribution over words. This approach estimates a topic from a document collection for each position. To construct such a collection for a particular position, I propose a seeding algorithm that extracts a set of terms relevant to the social position. Coherence-based evaluation shows that the proposed approach learns significantly more coherent representations when compared with a relevance modelling baseline.

Third, I present a diversification approach based on the proposed framework. Diversification algorithms aim to return a result list for a search query that would potentially satisfy users with diverse information needs. I propose to identify social positions that are relevant to a search query. These positions act as an implicit representation of the many possible interpretations of the search query. Then, relevant positions are provided to a diversification technique that proportionally diversifies results based on each social position's importance. I evaluate my approach using four test collections provided by the diversity task of the Text REtrieval Conference (TREC) web tracks for 2009, 2010, 2011, and 2012. Results demonstrate that my proposed diversification approach is effective and provides statistically significant improvements over various implicit diversification approaches.

Fourth, I introduce a session-based search system under the framework of learning to rank. Such a system aims to improve the retrieval performance for a search query using previous user interactions during the search session. I present a method to match a search session to its most relevant social positions based on the session's interaction data. I then suggest identifying related sessions from query logs that are likely to be issued by users with similar information needs. Novel learning features are then estimated from the session's social positions, related sessions, and interaction data. I evaluate the proposed system using four test collections from the TREC session track. This approach achieves state-of-the-art results compared with effective session-based search systems. I demonstrate that such a strong performance is mainly attributed to features that are derived from social positions' data.

Acknowledgements

I would like to express my eternal gratitude to my supervisor, Stephen Clark. This thesis would not have been possible without his continuous support. He taught me countless lessons with patience and gave me the freedom to explore various research questions. I have been extremely fortunate to learn from him. I also want to thank Lise Gough, the graduate education manager, and system administrators for their help on various occasions. My thanks are also due to members of the natural language and information processing group for creating an intellectually stimulating environment. I must also thank Andreas Vlachos and Leif Azzopardi for their valuable feedback during my PhD viva. I am also grateful to King Saud University for their generous scholarship.

I am sincerely thankful to my parents, brothers and sisters for their profound moral support. Finally, I would like to thank my wife, Alanoud, for her steadfast support and understanding, and my daughters, Lama and Deema, for being part of my life.

Contents

CHAPTER 1

Introduction

In 1950, Calvin Mooers coined the term *Information Retrieval* (IR) to refer to a system that uses punch cards to search and retrieve information (Mooers, 1950). Mooers's research and others prior to the 1950's (Sanderson and Croft, 2012) are all probably driven by a simple motivation. That is, a growing amount of information is being generated for which instant and accurate access is imperative yet increasingly difficult (van Rijsbergen, 1979). In recent years, information within a variety of mediums such as webpages, documents, images, and videos has continued to grow rapidly. As a result, the necessity and usefulness of IR tools have become even more apparent. The study of IR as a field of science comprises several topics that are best described in the following definition by Gerard Salton:

> Information retrieval is a field concerned with the structure, analysis, organization, storage, searching and retrieval of information (Salton, 1968).

A major repository of information and an area for IR research and commercial applications is the web. Web search engines are IR tools that enable prompt access to information that is contained on the web. They are perhaps the leading example of successful IR tools that form part of daily activities for most web users (Purcell et al., 2012). Their use has evolved from the primary goal of finding new pieces of information to tasks such as navigating the web (Broder, 2002) or re-finding information (Teevan et al., 2007a). However, the sheer number of webpages and users presents several technical challenges in all areas of IR, ranging from crawling and indexing to matching and ranking. The development of user-centric IR approaches for web search is the topic of this thesis.

A prominent approach to studying and developing IR tools, in general, is based on a systematic, laboratory-based, view of the field. In system-based IR, real users and their search tasks are abstracted away (Ingwersen and Järvelin, 2005; Kelly, 2009). The main rationale for such a viewpoint is to simplify the evaluation of retrieval models and other components. Researchers would need access to a test collection and experiments could be performed in laboratory settings without real users. The relevance of a document to a query is considered topical and objectively assessed by experienced judges (Ingwersen and

Järvelin, 2005). Whilst such a view brings several benefits such as the rapid development of IR methods and the scientific comparison between competing approaches (Sanderson, 2010), it has been challenged for a number of reasons. Firstly, a real user judges the relevance of a document to their information need subjectively and dynamically (Ingwersen and Järvelin, 2005; Teevan et al., 2010). Secondly, users communicate their information needs via natural language queries. Empirical evidence suggests that search queries are typically short (Jansen et al., 2000; Pass et al., 2006) and inherit linguistic ambiguity (Hafernik and Jansen, 2013; Sanderson, 2008; Song et al., 2009). These findings suggest that a search query is not a precise representation of a user's information need. The laboratory-based view replaces users with queries while search engines would potentially benefit from a view that considers users as an internal entity, rather than an external one, in the design of an IR system.

The need for such a design has long been recognised in the IR community (Allan et al., 2003; Teevan et al., 2010). Users turn to an IR system as one tool that supports their wider information seeking activities (Ingwersen and Järvelin, 2005). The understanding of each user's context, search tasks, and interests is integral in helping individuals to satisfy their information needs (Callan et al., 2007). At the core of this user-oriented IR is the computational representation of a user known as a user model. The construction of such a model relies heavily on the collection of users' data. Previous research gathered users' data via two methods: explicit and implicit. In the first approach, users are asked to provide information about their interests (e.g. Kelly et al., 2005; Liu et al., 2004). The alternative is to record users' search activities and transform it to a user model (e.g. Ge et al., 2018; Matthijs and Radlinski, 2011; Teevan et al., 2005).

This thesis presents a novel framework to model search engine users. It proposes a different data collection process than previous approaches. Unlike most personalisation approaches, the proposed framework is designed to preserve the privacy of users. User models are estimated from the web without real users' data. To accomplish such a goal, I build on role theory from the social science literature (Biddle, 1986; Linton, 1936). In this theory, people are members of different social positions. Examples of such positions might be *professor, student, traveller or football fan.* For each social position, a set of distinguishable behavioural patterns can be identified and used to associate the behaviour of a person to a particular position. This thesis suggests that both social positions and their behavioural patterns can be uncovered from the unstructured text on the web. A search engine user can then be represented based on their membership to one or more social positions. In this thesis, I present methods to identify and subsequently represent social positions from publicly available webpages. I also thoroughly evaluate the proposed framework on two extrinsic web search tasks: search results diversification and session-based search.

1.1 Research questions

In this thesis, I aim to address the following five main research questions:

RQ1. How can we build generalisable and privacy-preserving models of search engine users independent from real user interaction?

RQ2. Based on social role theory, how can we source and identify social positions in a dynamic and minimally supervised manner?

RQ3. Given a set of social positions, how can we learn a computational representation for each one of them that can be integrated into different retrieval applications?

RQ4. Can models of social positions be used to diversify search results?

RQ5. Assuming a search session, how can we improve retrieval performance for the user's next query using models of social positions?

1.2 Contributions

I make the following contributions:

1. In chapter 3, I introduce a user modelling framework based on role theory as an answer to my first main research question *RQ1.* A central assumption of the proposed framework is the use of public webpages as a resource to learn about users' interests. This assumption constitutes a departure from previous modelling approaches that consider users' search activities as the main resource to construct user models. Instead, the proposed framework builds on two concepts from role theory: a social position and a social role. A social role is a set of behavioural patterns that identify the behaviour of a person occupying a particular social position. In the proposed framework, a social position is a label for a group of users who are similar in search activities. The proposed framework consists of three main phases: social positions identification, representation learning, and matching. The first component is concerned with identifying social positions from web documents. These positions act as labels for user models that are estimated in the second phase. The third component focuses on matching a search query, session, or document to its most relevant social positions.

2. In chapter 3, I propose a method to identify social positions from webpages. I formulate this task as a binary classification task and introduce a set of linguistic patterns used to enumerate candidate social positions. My experiments indicate that

social positions can be identified with good accuracy from public and noisy sentences that are extracted from the web. This chapter addresses my second main research question (*RQ2*).

3. In chapter 4, I propose an approach to building a user model, a representation, for each social position in response to the research question *RQ3*. The proposed approach uses a novel topic model to represent each social position as a multinomial distribution over words. I also present a modular process to construct a document collection for each social position. These document collections are used to infer the user model for each social position. Evaluation based on coherence measures shows that the proposed model generates topics that are more coherent than a relevance modelling baseline.

4. In chapter 5, I validate the proposed framework on the task of search results diversification. A diversification approach aims to present a result list that could potentially satisfy a diverse set of information needs. This task represents a suitable extrinsic evaluation exercise for the proposed framework presented in chapter 3 and the models in chapter 4. Diversification is primarily motivated by the challenge of search query ambiguity. Users submitting an identical search query could be seeking to satisfy different information needs. Previous approaches have required an explicit representation of the possible information needs behind a search query. These are usually in the form of reformulated queries and are typically extracted from query logs. Alternatively, implicit approaches use a clustering algorithm to identify candidate subtopics for search queries. I present an implicit diversification approach based on social positions. Firstly, I match a search query to its most relevant social positions, then the matched social positions for each query form the candidate subtopics. Secondly, I propose a diversification strategy that diversifies a result list proportionally to a query's social positions. I thoroughly evaluate the proposed approach using the Text REtrieval Conference (TREC) test collections and compare it to various clustering techniques. Results show that diversification based on social positions improves the diversity of search results under various evaluation metrics. The proposed approach effectively outperforms implicit diversification baselines. This chapter provides an answer to the research question *RQ4*.

5. In chapter 6, I use the task of session-based search to extrinsically validate the proposed framework. The goal of session-based search is to utilise a user's interaction data to improve relevance for the user's next query at the session level. In this chapter, I formulate session search as a personalisation task under the framework of learning to rank. A user's interaction data is used to map the user to the most relevant social positions. The pre-computed models that are estimated in chapter 4

are then used to define novel learning features and extract related sessions from query logs. I use the identified related sessions to define further learning features for the learning to rank model. Experiments on TREC test collections demonstrate the effectiveness of the proposed approach compared with state-of-the-art session-based systems. This chapter addresses my fifth research question (*RQ5*) and provides the basis for the following publication:

- Aloteibi, S., & Clark, S. (2020). Learning to Personalize for Web Search Sessions. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 15-24.

CHAPTER 2

Background

The goal of an IR system is to retrieve *relevant* information to satisfy an information need (Manning et al., 2008). Relevance is, perhaps, an intangible notion (Büttcher et al., 2010) and a large body of research has explored its definition and factors that make a piece of information relevant to a user's information need (Saracevic, 2007). Systematically, the IR component that is responsible for assessing the relevance of a document to a search query is called a retrieval model. A retrieval model is a mathematical representation of a relevance theory (Croft et al., 2010). The effectiveness of such models is measured based on their performance on human annotated datasets. In this chapter, I explore general retrieval models as well as personalised approaches and conclude with an overview of evaluation approaches and metrics for such models. The aim is to provide a contextual background to IR fields that are relevant to the research presented in this thesis.

2.1 Early retrieval models

Retrieval models are generally classified based on their representation of documents and queries (Baeza-Yates and Ribeiro-Neto, 2011). In early IR research, two classes of models were at the centre of development. The first is the set theoretic approach, which represents a document and a query as sets of terms. The Boolean retrieval model is an intuitive example of set theoretic models. The relevance theory behind such a model is simple; that a document is either relevant or not relevant to a query that is expressed using Boolean logic. This model relies on the user firstly composing a precise query and then examining a potentially large number of documents that match the Boolean expression. In practice, not all users are willing to spend a significant amount of their time in examining results or formatting a complicated query. Furthermore, Boolean retrieval systems do not rank matched documents based on their relevance to the user query.

In the second class of models, representation is based on vectors in a t-dimensional space where t is the number of index terms. Such models are usually referred to as algebraic models. A prominent model in this category is the Vector Space Model (VSM)

(Salton et al., 1975). Let $V = \{t_1, t_2, \ldots, t_T\}$ be the set of terms occurring in a document collection. The VSM represents a document d as a weight vector of index terms as follows:

$$\vec{d} = (w_1, w_2, \ldots, w_T) \tag{2.1}$$

where w_n is the weight of term t_n in document d. The weight can simply be the number of occurrences of term t_n in document d. The query vector q is constructed in a similar way. Typically, a document-term matrix is constructed to represent the document collection. This representation enables computation of the similarity between the document vector d and the query vector q using a similarity measure. The similarity score can be interpreted as an estimate of the relevance of document d to query q. The outcome of measuring all documents' vectors to a query is a ranked list of documents based on their similarity score. Various similarity measures have been investigated with the cosine similarity being the most successful (Croft et al., 2010). The VSM forms the basis for the Latent Semantic Indexing model (LSI) reviewed in chapter 4.

In early IR research, a significant amount of research was devoted to the area of term weighting. The term-document relation is not necessarily a Boolean relation and it is practical to assume that some terms are more important to a document than others. Two properties have been extensively used to weight terms and subsequently rank documents. These are: term frequency (tf) and inverse document frequency (idf) (Robertson and Spärck-Jones, 1994). The term frequency is the number of occurrences of term t in document d. The assumption is that: document d is likely to be relevant to term t if term t occurs often enough in document d (Luhn, 1957). The inverse document frequency measures a term's specificity (Spärck-Jones, 1972) and it is defined as follows:

$$IDF(t) = log\frac{N}{n_t} \tag{2.2}$$

where N is the number of documents in the collection and n_t is the number of documents in which term t occurs. This weighting scheme values less frequent terms over frequent ones. Also, a variable relating to document length is usually incorporated (Robertson and Spärck-Jones, 1994). Assuming that term t has the same number of occurrences in two documents, the shorter document is more likely to be relevant than the longer one. These term weighting approaches and various others are continually used to weight terms in algebraic models, probabilistic models, and others.

2.2 Probabilistic models

Two concepts are at the centre of probabilistic retrieval models. The first is the *basic question* which asks the following (Spärck-Jones et al., 2000):

What is the probability that this document is relevant to this query?

The second concept is the Probability Ranking Principle (PRP) which states that (Robertson, 1977):

> If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its user will be the best that is obtainable on the basis of that data.

In probabilistic notations, the basic question seeks to estimate the probability of usefulness, or relevance, of document d to query q which can be written as follows (Lafferty and Zhai, 2003):

$$P(R = 1|d, q) = 1 - P(R = 0|d, q) \tag{2.3}$$

where R is a relevance random variable that takes two values (1 to denote relevant document and 0 for non-relevant). Following the PRP, the probability of relevance for each document in the collection needs to be estimated and then results are ranked accordingly. In such a framework, retrieval can be formulated as a binary classification task. For each query, there are two sets of documents in the collection: relevant and non-relevant (Baeza-Yates and Ribeiro-Neto, 2011; Croft et al., 2010).

The Binary Independence Model (BIM) makes some simplifying assumptions that enable estimation of equation 2.3. The model is binary because documents and queries are represented as binary term incidence vectors $\vec{d}$,$\vec{q}$. Terms are assumed to appear in a document independently of others. Since the goal is to obtain a ranking of documents in a decreasing order of relevance, the BIM computes the *odds of relevance* as follows (Manning et al., 2008):

$$O(R|\vec{d}, \vec{q}) = \frac{P(R = 1|\vec{d}, \vec{q})}{P(R = 0|\vec{d}, \vec{q})} \tag{2.4}$$

Equation 2.4 can be simplified to the following scoring formula:

$$\sum_{t \in q \wedge t \in d} log \frac{p_t(1 - s_t)}{s_t(1 - p_t)} \tag{2.5}$$

p_t denotes the probability that term t occurs in a relevant document and s_t is the probability that it appears in a non-relevant document. In the absence of information about relevant documents and non-relevant documents, p_t is assumed to be constant and

s_t is estimated based on the term's frequency in the document collection. BIM can then be written as follows:

$$\sum_{t \in q \wedge t \in d} log \frac{N - n_t + 0.5}{n_t + 0.5} \tag{2.6}$$

where N is the number of documents in the collection and n_t is the number of documents in which term t occurs. It is worth noting that this equation takes into account the IDF score of the terms but no other statistics such as the term frequency in the document or the document length. These statistics are included in the well-known Best Match model, which is commonly referred to as the BM25 (Robertson and Zaragoza, 2009; Robertson and Walker, 1994). The BM25 model has been shown to be useful empirically and continues to be used within various settings. There are a number of forms for the BM25 scoring function. The BM25 ranking function that is used in this thesis is as follows:

$$\sum_{t \in q \wedge t \in d} B_{t,d,q} \times log \frac{N - n_t + 0.5}{n_t + 0.5} \tag{2.7}$$

$$B_{t,d,q} = \frac{(k_1 + 1) \times tf(t,d)}{k_1 \left((1-b) + b \frac{len(d)}{Z} \right) + tf(t,d)} \times \frac{(k_3 + 1) \times tf(t,q)}{tf(t,q) + k3} \tag{2.8}$$

Z is the average document length in the collection and $k1$, b and $k3$ are the model's parameters that are usually empirically tuned. The values that are used in this thesis are: $k1 = 1.2$, $k3 = 8$, and $b = 0.75$. These values were found to be effective in TREC experiments (Croft et al., 2010; Robertson and Walker, 1994; Robertson et al., 1994). They are also the default values as in Terrier IR platform (Ounis et al., 2005).

2.3 Ranking based on language modelling

The goal of a language model is to assign probabilities to sequences of words. A particular type of such models called the unigram language model is often used in IR applications. In a unigram model, each word is assigned a score to represent the probability of observing it if we draw a word at random from the document collection. Applications of language modelling are rooted in many natural language processing tasks such as speech recognition, machine translation, and spelling correction. In IR, models that are based on language modelling use a document to estimate a generative model for queries (Büttcher et al., 2010). Given a query q and a document d, the task is to estimate the probability that the document's language model will generate the query, i.e. $P(q|d)$. Unlike probabilistic models where relevance is directly accounted for, language modelling makes an implicit assumption about relevance. A ranking function based on language modelling will order documents based on their probability of generating the query $P(q|d)$. This is often referred to as topical relevance.

The Query Likelihood (QL) model (Ponte and Croft, 1998) is a prominent example of such models. QL estimates a language model for each document in the collection and ranks documents based on the likelihood that the document is relevant to the query $P(d|q)$. Using Bayes' Rule, we get:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \tag{2.9}$$

Both $P(d)$ and $P(q)$ can be ignored since $P(d)$ is assumed to be uniform and $P(q)$ is constant for all documents. Equation 2.9 can therefore be re-written as:

$$P(d|q) \overset{rank}{=} P(q|d) \tag{2.10}$$

Assuming the document language model is a unigram and using Maximum Likelihood Estimation (MLE), we get:

$$P(q|d) = \prod_{i=1}^{n} P(q_i|d) = \prod_{i=1}^{n} \frac{tf(q_i, d)}{|d|} \tag{2.11}$$

A classic problem with equation 2.11 is that $P(q|d)$ will be zero for documents that are missing one of the query's terms. To overcome such data sparsity issue, unseen terms are assigned probability estimates that are subtracted from seen terms' probabilities, i.e. smoothing. A query term q_i probability of occurrence in document d can be written as:

$$P(q_i|d) = (1 - \alpha_d)P(q_i|d) + \alpha_d P(q_i|C) \tag{2.12}$$

where $P(q_i|C)$ is the probability of query term q_i occurring in the document collection C. A common method of smoothing utilises a Dirichlet prior given that it is the conjugate prior of the multinomial distribution (Zhai and Lafferty, 2004). The interpolating parameter α_d is set to be dependent on the document length as follows:

$$\alpha_d = \frac{\mu}{|d| + \mu} \tag{2.13}$$

μ is set empirically. The QL model can then be written as:

$$logP(q|d) = \sum_{i=1}^{n} log \frac{tf(q_i, d) + \mu \frac{tf(q_i, C)}{|C|}}{|d| + \mu} \tag{2.14}$$

Note that this language model is essentially a mixture of two multinomial distributions. The original work by Ponte and Croft (1998) follows a multi-variate Bernoulli query generation process that makes similar assumptions as the Binary Independent Model. Terms were assumed to be independent and the query is represented using a binary term incidence vector (Azzopardi, 2005; Liu and Croft, 2005). However, the multinomial process has become the standard for language modelling approaches (Baeza-Yates and Ribeiro-Neto, 2011). Other well known approaches in this area include the work of Miller et al. (1999) and Hiemstra (1998).

2.4 Relevance modelling

Retrieval models might have access to additional information about relevant and non-relevant documents in some search scenarios. For example, the user might provide explicit relevance feedback to the search engine by labelling documents as relevant or non-relevant. Alternatively, a user's judgment about relevance might be implicitly inferred based on the user's behaviour. In either case, the retrieval model should be able to accommodate such information. As discussed earlier, only the probabilistic models explicitly account for information about relevant and non-relevant documents. Additional modifications are needed for both the VSM and language modelling approaches to account for relevance information. One of the early and most widely used approaches in this domain is the Rocchio algorithm (Rocchio, 1971) which modifies the query vector q in the VSM to re-weight a query's terms based on their occurrences in relevant and non-relevant documents. In its most common form, the new query vector $\hat{q}$ is computed as follows (Croft et al., 2010):

$$\hat{q}_t = \alpha \times q_t + \beta \frac{1}{|R|} \sum_{d_i \in R} d_{i,t} - \gamma \frac{1}{|N|} \sum_{d_i \in N} d_{i,t} \tag{2.15}$$

where R and N are the sets of relevant and non-relevant documents, respectively. q_t is the weight for term t in the original query vector q. $d_{i,t}$ is the weight for term t in document i.

Lavrenko and Croft (2001) presented methods to estimate a relevance model within the language modelling framework. A relevance model is a language model for words in relevant documents. The main challenge in constructing such a model is that labelled documents are typically not available. If such training data are available, then a relevance model can be easily estimated using MLE. The assumption behind Lavrenko and Croft's model is that query terms and relevant documents are both sampled from a relevance model R. The probability of observing a word w is conditioned on observing the query terms $q_1 \ldots q_n$. Formally, we can write the following:

$$P(w|R) \approx P(w|q_1, \ldots, q_n) \tag{2.16}$$

The joint probability of the new word w and the query terms $q_1 \ldots q_n$ is estimated as follows:

$$P(w, q_1, \ldots, q_n) = \sum_{d \in D} P(d) P(w|d) \times \prod_{i=1}^{n} P(q_i|d) \tag{2.17}$$

D is the set of documents that are assumed, i.e. a pseudo-relevance assumption, or known to be relevant. Documents are represented by their multinomial language models. $P(d)$ is the document prior, which is set uniformly for all documents. Note that $\prod_{i=1}^{n} P(q_i|d)$ is the QL score of document d. The ranking of documents is then performed by computing the Kullback-Leibler divergence score between the two language models: the relevance and the document models. Throughout this thesis, I use this instantiation of Lavrenko and Croft's models.

2.5 Learning to rank

Modern search engines have access to a number of relevance indicators. Some of these indicators could be just the scores of traditional retrieval models as reviewed in the previous sections. Others could be based on users' behaviour or user-specific indicators, e.g. previous visits to a particular webpage. Rather than relying on a single and manually tuned retrieval model, machine learning techniques could be used to *learn* a retrieval model that effectively combines multiple relevance indicators, or features. In this section, I review a class of retrieval models based on the framework of learning to rank (LTR). Specifically, I focus on the particular case of learning to rank approaches with labelled training data in the form of relevance judgments.

LTR is a discriminative learning task. There are four key elements to LTR approaches (Liu, 2009). The first is the input space which represents the objects on which an LTR algorithm is to be applied. In the context of this thesis, objects are strictly web documents. Each document d is represented by a feature vector $x_d \in R^d$. Features could represent various relevance or quality indicators about the document. The second element is the output space to which a document feature vector x_d is mapped using a ranking function. This function belongs to the third component, which is the hypothesis space. The fourth component is the loss function which quantifies the discrepancy between the ranking function output and the true output as provided in the format of relevance judgments. The goal is to learn a ranking function that best resembles human relevance judgments.

It is common to categorise LTR approaches into three classes: pointwise, pairwise, and listwise. In pointwise approaches, ranking is formulated as a classification, regression, or ordinal regression problem. The output space is either non-ordered categories, real values, or ordered categories, respectively. The loss function is computed on the basis of a single document. Whilst such a definition is widely used in other machine learning tasks, it is problematic in ranking applications (Liu, 2009). Firstly, pointwise approaches do not account for a document's position in the ranked list; a property that is integral to a number of widely used evaluation metrics, e.g. nDCG and nERR. Secondly, relative ordering between documents is not considered during learning despite the fact that ranking is more concerned with relative ordering than predicting relevance scores. Examples of this category include PRanking (Crammer and Singer, 2002) and McRank (Li et al., 2007).

Pairwise methods model preferences between pairs of documents rather than attempting to assign a relevance score to each document independently of others. In the terms of Liu's (2009) four components of LTR, the input space is a pair of features vectors x_i and x_j which represent documents d_i and d_j. The output space is 1 if $d_i \prec d_j$ and -1 otherwise for a specific query q. Note that pairwise approaches still do not account for the document's position in the final ranked list but merely the relative order between a pair of documents. RankNet (Burges et al., 2005), RankBoost (Freund et al., 2003), and Ranking SVM (Joachims, 2002) are some examples of pairwise methods.

For listwise approaches, the input space includes all of the feature vectors associated with all documents for a specific query. These approaches attempt to optimise an IR evaluation metric which is a challenging task because such metrics are often not continuous. Alternatively, ground truth data could be provided as the *gold* permutation of documents for a specific query. The loss function can then be defined based on the list of documents produced by the ranking function and the ground truth list. Depending on the loss function, the output space could be a relevance score for each document or a permutation of documents. Examples of such approaches include ListNet (Xia et al., 2008) and AdaRank (Xu and Li, 2007).

2.5.1 RankNet, LambdaRank, and LambdaMART

In chapter 6, I use LambdaMART (Burges, 2010; Wu et al., 2010) to re-rank documents for the session search task. To better discuss LambdaMART, RankNet and LambdaRank need to be introduced. RankNet (Burges et al., 2005) is a pairwise approach that uses a neural network as the underlying scoring function F. Feature vectors x_i and x_j are scored using F resulting in the scores s_i and s_j, respectively. These two scores are mapped to a learned probability $P_{i,j}$ which denotes that document i should be ranked higher than document j, i.e. $d_i \prec d_j$. Mapping is performed using the sigmoid function:

$$P_{i,j} \equiv \frac{1}{1 + e^{-\sigma(s_i - s_j)}} \tag{2.18}$$

The cost function is defined based on the cross entropy between the modelled proability

$P_{i,j}$ and the ground truth probability $\widehat{P_{i,j}}$ as follows:

$$C = -\,\widehat{P_{i,j}}\, log\, P_{i,j} \;-\; (\,1 \;-\; \widehat{P_{i,j}})\, log\,(1 \;-\; P_{i,j}) \tag{2.19}$$

RankNet uses gradient descent to learn the scoring function F. LambdaRank is based on RankNet with two main differences. Firstly, the derivatives are defined *after* sorting the documents based on their current model scores. Such derivatives of the cost with respect to the model scores are called $\lambda-$gradient. The motivation for such a decision is that models are trained to minimise an optimisation cost, e.g. cross entropy, but such a process may not lead to improvement when evaluating ranked lists of documents using an IR metric. This approach allows for optimisation of an IR metric because the gradient will depend on the change that might result when the positions of the two documents under investigation are swapped. A bigger gradient would mean a bigger impact on the IR metric. Secondly, training algorithms, e.g. neural networks, only require the $\lambda-$gradient and not the direct cost. The $\lambda-$gradient for LambdaRank is simply the same as for RankNet weighted by the size of the change in an evaluation metric as in equation 2.20.

$$\lambda_{LambdaRank} = \lambda_{RankNet} \times |\bigtriangleup IR - metric| \tag{2.20}$$

$|\bigtriangleup IR - metric|$ could be based on nDCG, nERR or any other suitable evaluation metric. The $\lambda-$gradient can be considered as a force that moves a particular document i up or down the ranked list (Burges, 2010). In RankNet, the $\lambda-$gradient is calculated for each particular document i as follows:

$$\lambda_i = \sum_{j:\{i,j\}\in I} \lambda_{ij} - \sum_{j:\{j,i\}\in I} \lambda_{ij} \tag{2.21}$$

where I is the set of pairs of documents such that $i, j \in I$ indicates that $d_i \prec d_j$. The $\lambda-$gradient for a particular pair of documents is calculated as:

$$\lambda_{i,j} = \frac{\partial C(s_i - s_j)}{\partial s_i} \tag{2.22}$$

Note that s_i is the score that the model assigns to document d_i. LambdaMART is based on Multiple Additive Regression Trees (MART) (Friedman, 2001) and LambdaRank. MART is a boosted tree model in which the model is a linear combination of a set of regression trees. MART's output can be written as follows (Burges, 2010):

$$F_N(x) = \sum_{i=1}^{N} \alpha_i f_i(x) \tag{2.23}$$

where N is the number of trees in the ensemble. $f_i(x) \in R$ is a regression tree model and α_i is the weight of regression tree i. The output of each tree is a fixed value γ_{kn}

Algorithm 1: LambdaMART (Burges, 2010).

```
Input: N: number of trees.
m: number of training samples.
L: number of leaves per tree.
$\eta$: learning rate.
begin
    foreach $i = 1...m$ do
        $F_0(x_i) = BaseModel(x_i)$   // Set to 0 for empty base model.
    end
    foreach $k = 1...N$ do
        foreach $i = 1...m$ do
            $y_i = \lambda_i$   // Calculate the $\lambda-$gradient.
            $w_i = \frac{\partial y_i}{\partial F_{k-1}(x_i)}$   //Derivative of $\lambda$-gradient for $x_i$.
        end
        $\{R_{lk}\}_{l=1}^{L}$   // Create L-leaf tree on $\{x_i, y_i\}_{i=1}^{m}$.
        $\gamma_{lk} = \frac{\sum_{x_i \in R_{lk}} y_i}{\sum_{x_i \in R_{lk}} w_i}$   // Assign leaf values based on Newton step.
        $F_k(x_i) = F_{k-1}(x_i) + \eta \sum_l \gamma_{lk} 1(x_i \in R_{lk})$
    end
end
```

associated with leaf k of tree n. MART applies the least squares cost function to find the splits and gradient descent to decrease the loss when training the next tree (Burges, 2010). LambdaMART uses MART with the cost function of LambdaRank. Leaf values are calculated using a Newton's approximation step. LambaMART is presented in algorithm 1.

As mentioned earlier, a learning to rank model would represent each document by a feature vector. These features encode information about the document itself and its interaction with a search query, e.g. relevance scores based on standard ranking models. Manual feature engineering is a time-consuming task that would potentially require manual updates. Recent advances in neural networks have led to many neural-based IR ranking approaches that learn such features from the data. Based on their architecture, these models are typically categorised into two groups: representation-focused and interaction-focused (Guo et al., 2016). Representation-focused models learn independent representations for query and document. The relevance of a document to a search query is then computed based on a matching function that takes their representations as inputs (e.g. Hu et al., 2014; Huang et al., 2013; Palangi et al., 2016; Shen et al., 2014). Interaction-focused approaches, on the other hand, learn a joint representation for search query and document. This representation is fed to a deep neural network to output a relevance score (e.g. Dai et al., 2018; Guo et al., 2016; Hui et al., 2017; Xiong et al., 2017). Some hybrid approaches have also been proposed in the literature to combine both architectures (Mitra et al., 2017; Wang et al., 2016). Guo et al. (2020) and Mitra et al. (2018) provide extensive surveys of neural-based IR models.

2.6 Personalisation

In previous sections, I reviewed standard retrieval models whereby an information need is represented by a search query. This representation means that if two or more users submit the same query then their information need is identical and the same result list will be returned for such users. In practice, such an assumption does not usually hold. Search queries are typically short (Jansen et al., 2000; Spink et al., 2001; Zhang and Moffat, 2006) and ambiguity of various types is commonly observed (Sanderson, 2008). These empirical observations have led to extensive research in the area of personalisation where an additional user representation is incorporated in the retrieval process to tailor search results for each user. A user representation is commonly referred to as a user profile or a user model. Gauch et al. (2007) described three different phases of personalisation; data collection, representation, and usage.

The first phase involves collecting information about users in order to represent them. Three key questions guide the data collection phase which are; how to define a user, what information to collect, and how to gather them. The first question determines the level of specificity of the user model to be built. A user model can be at the individual (Ge et al., 2018; Liu et al., 2004; Matthijs and Radlinski, 2011; Sieg et al., 2007; Sontag et al., 2012; Teevan et al., 2005; Zhou et al., 2020) or group level (Bennett et al., 2011; Mei and Church, 2008; Rich, 1979; Shapira et al., 1997; Smyth, 2007; Teevan et al., 2009; Zhao et al., 2014). A user's information is then converted to a representation in the second phase of personalisation. This second step involves choosing a suitable data structure and sometimes an update mechanism for users' profiles to account for the dynamic nature of users' interests such as a spreading activation algorithm (Sieg et al., 2007), exponential decay (Vu et al., 2015; White et al., 2010) or re-weighting long-term profiles based on session-level interaction (Ge et al., 2018). The third component addresses the question of how to integrate the user model into the retrieval process. A common strategy to employ user models is re-ranking where an initial ranked list is processed based on the current user's interest. Alternatives to the re-ranking strategy include query expansion using terms that are extracted from the user's profile (Chirita et al., 2007) or integrating the user's model directly into the retrieval model (Harvey et al., 2013). Ghorab et al. (2013) and Liu et al. (2020) provide detailed surveys of personalised IR. Group-based approaches are discussed in the next chapter.

There are two modes of collecting users' information: explicit and implicit. Explicit approaches rely on users directly providing the system with additional information about their interests, information need, or direct relevance judgements. Various approaches have been introduced in the literature including users answering a short survey about the search topic (Kelly et al., 2005), selecting additional expansion terms (Belkin et al., 2005), specifying the search task via a re-design of the user interface (Ahn et al., 2008), and selecting topical categories of interests (Liu et al., 2004). On the other hand, implicit approaches rely on users' behaviour and usage to collect information about their interests. Implicit approaches predominate explicit ones (e.g. Cai et al., 2014; Ge et al., 2018; Li et al., 2014; Matthijs and Radlinski, 2011; Sontag et al., 2012; Teevan et al., 2005; Vu et al., 2017) due to their unobtrusive nature, ease of collection, users' reluctance in providing information about their interests, and occasional inconsistency in provided information (Carroll and Rosson, 1987; Gauch et al., 2007). Additionally, research in the area of relevance feedback has found that implicitly inferred expansion terms perform as well as those provided by users (Belkin et al., 2005; Teevan et al., 2005). Some methods might combine the two approaches (Liu et al., 2004; Psarras and Jose, 2006). For example, Liu et al. (2004) asked users to select the topical category of their query from a few candidate categories selected based on their browsing behaviour.

Various data sources and structures have been explored to build and represent users' profiles. Teevan et al. (2005) built a client-side index of several sources of information such as documents created by the user, visited webpages, emails, and calendar items. They used this local index as a representation of the user to re-rank documents returned by a search engine using the BM25 model which naturally accounts for relevance feedback. Matthijs and Radlinski (2011) constructed a vector of weighted terms for each user based on the content of webpages that the user had visited. Another predominant approach to building a user profile is to infer each user's topics of interests. The topic ontology of the Open Directory Project (ODP) has been used extensively to build user profiles in a number of personalisation approaches where an interest score or a probability estimate of the user's interest in a specific category is calculated based on the user's previous queries and clicked documents (e.g. Bennett et al., 2012; Chirita et al., 2005; Liu et al., 2004; Sieg et al., 2007; Sontag et al., 2012). However, maintaining such an ontology to ensure coverage of emerging topics and classifying documents accordingly is an expensive and tedious task (Vu et al., 2015). Alternatively, users' topical interests can be estimated using a statistical topic modelling approach such as the Latent Dirichlet Allocation (LDA) over the users' clicked documents (Carman et al., 2010; El-Arini et al., 2012; Harvey et al., 2013; Vu et al., 2015).

More recently, representations based on deep learning have been investigated to build fine-grained models of users (Li et al., 2014; Lu et al., 2019; Vu et al., 2017; Zhou et al., 2020). For example, Vu et al. (2017) represented a user by a user embedding that is learned based on the user's previous queries and clicked documents at the session level. Ge et al. (2018) generated user profiles based on recurrent neural networks from the user's long-term interests. A query-aware attention model is also constructed based on the user's current session and used to weight the long-term interests. Lu et al. (2019) applies generative adversarial networks to generate discriminative negative examples to build fine-grained user models.

2.7 Evaluation

An IR system can be evaluated based on several dimensions such as its usability, cost, efficiency, and effectiveness (Sanderson, 2010). The latter has received considerable attention in the community and continues to be a key evaluation criterion. The effectiveness of a retrieval system is primarily relative to the user's information need. After all, the purpose of an IR system is to locate and return relevant information to its users. Measuring effectiveness also depends on the assumptions that are made about users and search queries. A system that follows the Probability Ranking Principle would assume a single information need and seek to order documents in descending order of relevance. In another scenario, such as diversification, a different assumption about the search query is made and thus it requires a different evaluation methodology. In this section, I discuss strategies to evaluate and measure effectiveness of IR systems.

2.7.1 Evaluation strategies

Evaluation strategies can be categorised into two classes: system-based and user-based (Kelly, 2009; Voorhees, 2001). System-based approaches focus on the relevance of the returned results whilst user-based evaluate whether users can use the system to fulfil an information need (Kelly, 2009). User-based evaluation encompasses system-based approaches in addition to studying users' information seeking behaviour and interaction with the system and information (Kelly, 2009) as well as their satisfaction (Voorhees, 2001). Although conducting such studies results in qualitative and user-centric findings that are difficult to synthesise in system-based methods, they are often costly and difficult to perform and scale.

Relevance judgments need to be collected in order to perform system-based evaluation. Users could be explicitly asked to judge the relevance of documents in user studies settings (Liu et al., 2004; Teevan et al., 2005). These studies, however, typically involve small numbers of participants. Alternatively, implicit indications of relevance could be used to gather relevance labels. The primary indication occurs when a user clicks on a result and remains on the clicked document for a while. This implicit feedback does not intervene with users' typical search activities. These are mostly gathered from query logs and the last document or documents with a satisfying clickthrough are considered relevant to the query (Agichtein et al., 2006; Cai et al., 2014; Dou et al., 2007; Fox et al., 2005; Ge et al., 2018; Harvey et al., 2013; Joachims et al., 2005; Li et al., 2014; Vu et al., 2015). However, a number of biases are inherited with such a resource. For example, users tend to click on the top documents even if they are less relevant than lower documents and may click on weakly relevant documents if the quality of the ranked list is poor (Joachims et al., 2005).

Results from two competing retrieval systems could be presented side-by-side to enable users to explicitly select their preferred system (Thomas and Hawking, 2006). In such a setting, users would evaluate the entire ranked-list of documents once per query. Others have suggested presenting a single rank list that contains documents from two different ranking approaches. Each document comes from a single retrieval approach. This approach is known as interleaved evaluation where users' clickthrough decisions determine which ranking function is preferred (Joachims, 2002; Matthijs and Radlinski, 2011; Radlinski et al., 2008b). However, the development of a retrieval model requires various configuration and analysis steps which can be difficult and costly to reproduce when relevance judgments are sourced from real users. Another widely used approach involves relying on experts to produce queries and judge relevance of documents. This paradigm was first introduced by Cleverdon (1959) and is commonly referred to as the Cranfield approach or test collection based evaluation.

The use of test collections to evaluate IR systems has become the *de facto* standard of evaluation since its early conception (Sanderson, 2010). Such collections are typically developed as part of evaluation conferences. Examples include TREC the flagship of such conferences, NTCIR for Asian languages IR and CLEF for European languages IR. These conferences bring at least two important benefits to the IR community (Büttcher et al., 2010). Firstly, it encourages credible progress to be made in specific IR tasks through the provision of shared data enabling comparisons between participants' approaches. Secondly, it produces test collections that can be reused by researchers outside the scope of these conferences to validate their research.

Each test collection consists of three components (Sanderson, 2010). Firstly, a collection of documents on which retrieval experiments and evaluation is performed (often referred to as a document collection). Secondly, a set of *topics* which represent statements about users' needs and their associated *queries*. Thirdly, a set of relevance judgments for $< document, query >$ pairs (often referred to as *qrels*). A technique known as *pooling* is used to collect sets of candidate documents to be judged. Participants in an evaluation conference submit a list of candidate documents for each query, known as *runs*. The pool of candidate documents for each query is the union of top documents from each run for that query. Experts will then judge the relevance of a document to a query on a binary or graded scale.

2.7.2 Evaluation metrics

A number of evaluation metrics have been introduced to quantify effectiveness of a system in returning relevant documents to users' information needs. In this section, I present measures that assume a single information need for each query, i.e. follows the PRP.

Recall[k], P[k], AP and MAP: The most classical measures used to evaluate search systems are Recall[k], P[k], AP and MAP. The first stands for recall at rank cutoff k. The second is precision at k. The third refers to Average Precision and the latter to Mean Average Precision. Both Recall[k] and P[k] have been in use to evaluate effectiveness since the beginning of using test collections (Sanderson, 2010). They are defined as follows:

$$P[k] = \frac{\sum_{i=1}^{k} rel(d_i)}{k} \tag{2.24}$$

$$Recall[k] = \frac{\sum_{i=1}^{k} rel(d_i)}{R} \tag{2.25}$$

where $rel(d_i)$ is the binary relevance judgment of document d at rank i. R is the number of relevant documents in the collection. P[k] is simply the number of relevant documents normalised by the size of the ranked list, which is typically $k = 10$. Recall[k] is the fraction of relevant documents that are retrieved up to a rank cutoff k. The AP uses P[k] as follows:

$$AP = \frac{\sum_{i=1}^{N} (P[i] \times rel(d_i))}{R} \tag{2.26}$$

where N is the number of retrieved documents. Both R and N are query dependent. MAP is the mean AP of all test queries.

Expected Reciprocal Rank (ERR): Chapelle et al. (2009) introduced ERR as an adhoc evaluation metric based on the cascade user model. It discounts documents not only based on their position but also based on the relevance of the preceding documents. The more relevant the preceding documents are, then the lower the contribution that the relevance of the current document makes to the final score. ERR[K] is calculated as follows:

$$ERR[k] = \sum_{r=1}^{k} \frac{1}{r} \prod_{i=1}^{r-1} (1 - R_i) R_r \tag{2.27}$$

where R is a mapping function from relevance grade to relevance probability such that:

$$R(g) = \frac{2^g - 1}{2^{g_{max}}} \qquad \text{where } g \in \{0, 1, \ldots, g_{max}\} \text{ which is the graded relevance judgment.} \tag{2.28}$$

nERR[k] refers to the normalised version of ERR[k] as in the following definition:

$$nERR[k] = \frac{ERR[k]}{IERR[k]} \tag{2.29}$$

where IERR[k] is the ideal ERR[k] score that is computed over an ideal ranking of documents in decreasing order of relevance. Both ERR and nERR are calculated over the complete rank list when k is not identified.

Normalised Discounted Cumulative Gain (nDCG): An intuitive measure of effectiveness is to sum up the gain that the user accumulates from reading documents up until a specified rank (Järvelin and Kekäläinen, 2002). This measure is known as the Cumulative Gain (CG) and is defined as follows:

$$CG[k] = \sum_{i=1}^{k} rel(d_i) \tag{2.30}$$

where $rel(d_i)$ is the relevance of document d at rank i. The relevance score can be graded and not just a binary score. To emphasise the importance of ranking the most relevant documents at higher ranks, Järvelin and Kekäläinen (2002) suggested discounting the relevance of each document by the log of its rank (typically log base 2):

$$DCG[k] = rel(d_1) + \sum_{i=2}^{k} \frac{rel(d_i)}{log_2(i)} \tag{2.31}$$

Burges et al. (2005) used a modified version of DCG that rewards systems which allocate highly relevant documents at the top of the rank list. In chapter 6, I use this version of DCG. It is defined as follows:

$$DCG[k] = \sum_{i=1}^{k} \frac{2^{rel(d_i)} - 1}{log_2(i+1)} \tag{2.32}$$

nDCG[k] is calculated in a similar way as the nERR[k]. An IDCG[k] is calculated for an ideal rank list and then used to normalise DCG[k].

2.8 Summary

In this thesis, I propose a framework for learning user models from web documents. These models are typically used within a retrieval model to personalise the ranking of documents for each user. In this chapter, I reviewed different approaches to define retrieval models. In section 2.1, I discussed early models that belong to two categories: set theoretic and algebraic approaches. In sections 2.2 and 2.3, I introduced two families of models that are based on probabilistic foundations or language modelling, respectively. Early retrieval models and language modelling based approaches do not account for retrieval situations where implicit or explicit relevance feedback is provided. In section 2.4, I presented two popular approaches to estimate a relevance model from relevant, or pseudo-relevant, documents. An introduction to learning to rank models was provided in section 2.5 with a particular focus on RankNet, LambdaRank, and LambdaMART as examples of LTR approaches. In section 2.6, I discussed personalisation approaches with the goal of tailoring search results for each user using a computational model of the user. I reviewed personalisation research based on three main phases: data collection, representation and implementation. Finally, a review of different evaluation methodologies and metrics was provided in section 2.7.

CHAPTER 3

Role-based user modelling

3.1 Background

User modelling plays a pivotal role in research areas such as IR (Harvey et al., 2013; Sontag et al., 2012; Teevan et al., 2009; Vu et al., 2015), adaptive hypermedia (Brusilovsky, 2001) and recommendation systems (Pazzani and Billsus, 2007). Common to these fields is the abundance of information that stretches beyond the user's ability to process such information. Systems in such areas seek to optimise their user experience by filtering irrelevant information. This process is known as personalisation and it requires a model for each user because relevance is not universal among users. A user model contains information about the user that is supposed to help in judging the relevance of a piece of information to the user. Systematically, the model represents the user within the system. Thus, the construction and maintenance of such models is central to the effectiveness of personalisation algorithms.

In this chapter, I present a novel framework for building and maintaining user models for web search. Frameworks to model web users face three challenges. Firstly, search engines interact with millions of heterogeneous web users and personal information about each user must be collected. In the previous chapter, I explained that there are two methods for collecting user information: explicit and implicit. Explicit methods require users to provide information about their interests and to update their profiles accordingly. In contrast, the implicit approach uses information recorded during users' interactions with the search engine to model users. The sensitivity of such information represents the second challenge in building user models. Users' privacy concerns might not allow search engines to use interaction information to its fullest potential.

The third challenge relates to the broader objective of personalisation. Search engines should function as information assistants capable of predicting user needs even without queries (Allan et al., 2012) in addition to suggesting relevant and timely information. The goal is to enrich and simplify user interaction with a plethora of available information more than merely disambiguating queries. To achieve such a goal, a better understanding of users' information seeking behaviour is needed.

According to Malone et al. (1987), personalisation, or information filtering, can be performed based on a cognitive, social, or economic basis. Cognitive filtering uses content to model users. Textual content that is believed to be of interest to the user is transformed into representations such as terms vectors (Sugiyama et al., 2004), concept networks (Micarelli and Sciarrone, 2004), topic taxonomies (e.g. Bennett et al., 2012; Chirita et al., 2005; Liu et al., 2004; Sieg et al., 2007; Sontag et al., 2012), or topic models (e.g. Carman et al., 2010; Harvey et al., 2013; Vu et al., 2015) to form the user model. Hanani et al. (2001) expanded the cognitive approaches to include properties-based filtering alongside the content-based approaches. The properties-based filtering includes modelling of user goals, personality, topical interests, and other personal properties. Brusilovsky and Millán (2007) refer to this type of personalisation as feature-based, which includes features such as: user knowledge, interests, tasks, personality traits, and context.

The motive behind modelling such features about the user is that forming an understanding of individual users' information seeking behaviour or their personal characteristics cannot be achieved by modelling topical interests alone. An IR researcher and a marketer may both be interested in the topic of personalisation using machine learning techniques, but their goals and tasks may vary. A typical goal for a researcher is to be familiar with current technical details of their domain and to develop original solutions. In contrast, the marketer's goal would likely be to use the techniques to achieve higher success metrics for a campaign. Social approaches rely on sociological concepts to filter information such as the relevance of information to related people (Carmel et al., 2009) while economic filtering applies economic principles such as the potential gain versus the time cost of processing information.

Content-based methods are straightforward to implement. Documents that a user has interacted with (e.g. Matthijs and Radlinski, 2011) or stored locally on his/her machine (e.g. Teevan et al., 2005) are concatenated and then transformed into a suitable structure with some decay mechanism to adapt to users' change of interests. This process is best used to understand and organise *text* not necessarily *people* despite its wide use in personalisation as discussed previously. The first contribution of this chapter is a different process to model users. Each document is a source of information not just about a particular topic but also about human behaviour, tasks, and goals. For example, a document discussing a recent medical discovery to treat a particular disease not only communicates to the reader information about the new treatment but also that doctors might conduct medical experiments and patients can participate in medical trials. Such publicly available information on the web can collectively, and in conjunction with other

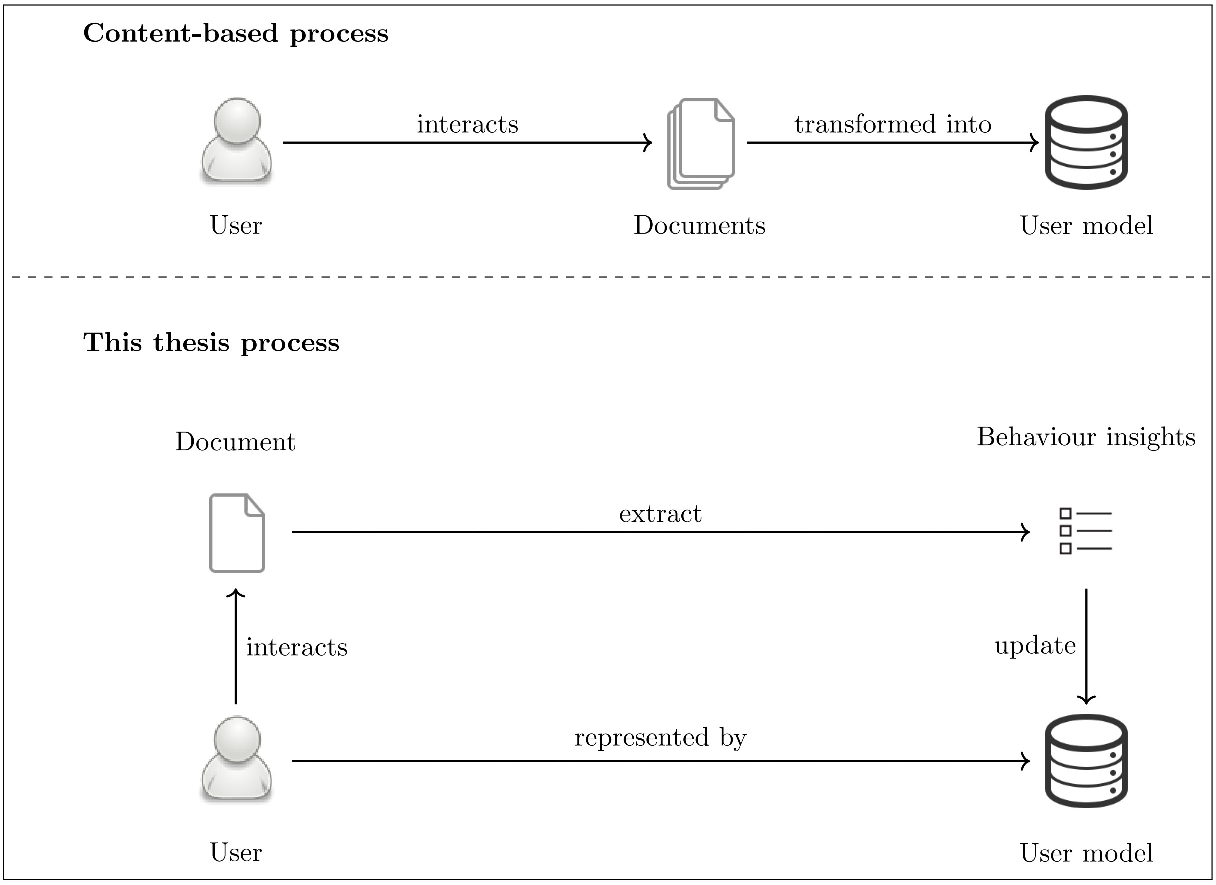


Figure 3.1: The difference between this thesis and content-based approaches for building a user model.

data[1], be viewed as a platform to understand, albeit partially, users and their information seeking behaviour. The patient in the previous example might want to search about benefits of participating in medical trials and the doctor may want to know about methods of recruiting patients.

The enablers for such an objective are the advances in Natural Language Processing (NLP) techniques. Researchers have already used NLP algorithms in areas such as sentiment analysis (Pang and Lee, 2004), identification of personality traits (Schwartz et al., 2013), the inferring of event chains (Chambers, 2011), or the building of personas (Bamman, 2015) which can all be viewed as aspects of human behaviour. I present a general process that builds on the assumption that each document should be considered as a source for insights about its intended audience's information needs and tasks. Users are represented by the behavioural insights extracted from documents they interact with rather than by the textual content of such documents. Figure 3.1 illustrates the proposed process in comparison with content-based modelling approaches.

The second contribution is a higher-order representation of users. It is natural to assume that users' interests, tasks, and goals are not independent from each other. For

[1] For example, query logs or clickthrough data.

example, a user goal to travel to London would be related to topical interests such as travel tips or London landmarks. This relatedness may extend to other tasks such as booking a hotel or applying for a visa. Similarly, intra-relatedness among tasks, topical interests, or goals is plausible. The process above suggests identifying the intended audience for each document and extracting behavioural insights at the indexing stage independently from any real user interaction. The identification of the intended audience is based on grouping related behavioural insights, which include topical interests and tasks, into a single cluster. Each cluster represents an audience.

This procedure is motivated by role theory, one of the theoretical perspectives in the social science literature. Biddle (2013, p. 4) defines role theory "*as a science concerned with the study of behaviors that are characteristic of persons within contexts and with various processes that presumably produce, explain, or are affected by those behaviors*". Linton (1936) discusses the importance of behavioural patterns in relation to the functioning of societies. People with a particular social position perform specific behavioural patterns (Linton, 1936). Biddle (1986) explained role theory through its resemblance to a theatrical show where actors in a play follow a script detailing their contribution to the overall performance. Examples of social positions include *student*, *mother*, *computer scientist*, *traveller*, or *Chelsea fan*. Within the context of social role theory, five main concepts are typically introduced and discussed. These are role, position, expectation, conformity, and function. These five concepts can be linked to five main propositions that underline role theory as stated by Biddle (2013, p. 8):

- Role theorists assert that "some" behaviors are patterned and are characteristic of persons within contexts (i.e., form roles).

- Roles are often associated with sets of persons who share a common identity (i.e., who constitute social positions).

- Persons are often aware of roles, and to some extent roles are governed by the fact of their awareness (i.e., by expectations).

- Roles persist, in part, because of their consequences (functions) and because they are often imbedded within larger social systems.

- Persons must be taught roles (i.e. must be socialized) and may find either joy or sorrow in the performances thereof.

The first two of these propositions are intertwined. They are concerned with the concepts of social role and social position, respectively. In the theatre example, each actor will be given a script that prescribes their performance during a play. Although it might not be written or even agreed upon, the script in life is the social role or the observed behaviour of a person occupying a particular social position. Various definitions have been introduced in the literature to define the key concept of social role. Linton (1936, p. 114) states that "*[a] role represents the dynamic aspect of a status*" where the notion of status is used similarly to that of positions. Biddle (2013, p. 58) defines "*a role to be those behaviours characteristic of one or more persons in a context*". Central to Biddle's definition is that a social role is context-specific and characteristic. For example, a teacher would teach in the context of a school class but not in a football game. Teaching is characteristic and observed behaviour of teachers but not football players, for example. A Social role is performed by persons who are typically given an identifiable label in society. This label is called a social position. A social position can be defined as "*an identity used for designating two or more persons who presumably share one or more overt characteristics*" where an identity is defined as "*a symbol that is used to designate one or more human beings*" Biddle (2013, p. 89 & 91).

An intriguing question arises from the above discussion. Why do specific persons have characteristic behaviour in certain contexts? For the teacher example, the answer might be that it is part of a teacher's job description to teach. However, the answer might not be evident for other social positions such as *mother*, *father*, *football fan*, and *tourist*. Role theorists posit that *shared expectations* among members of a social position govern their behaviour in a given context (Biddle, 2013). Similarly, a person will have established sets of expectations for people occupying certain social positions. These expectations for certain social positions are also likely to be held by other people in a specific society. An expectation is "*a statement that expresses a reaction about a characteristic of one or more persons*" (Biddle, 2013, p. 119) which may be conveyed overtly or covertly. These expectations could be formed by formal means such as laws, code of conduct, and contracts. Expectations could also be formed via observation of others behaviour. People would conform to these expectations for various possible reasons, such as their inner belief in them or to avoid sanctions, criticism, or social exclusion.

The concept of expectation is tightly linked to the concepts of conformity and sanctioning. Conformity indicates that people behave and expect others to behave in conformity to the expectations that they hold (Biddle, 2013). Sanctioning is one, perhaps weak, reason for a person to conform to role expectations because of their anticipation of sanctions that might be imposed on them by other compliant people. Finally, the function concept is due to a systematic view of society. Behaviours serve functions and have objective effects (Biddle, 2013). As I reviewed earlier, a role is a set of behaviours expected from a person occupying a social position. It is the integration of such roles, or functions, that results in a social system. It should be noted that there are debates about such key concepts of role theory in the social science literature. Biddle (2013) provides a comprehensive discussion of role theory and its main concepts.

In IR context, search engines are used to fulfil an information need. It is natural to assume that such information needs do not arise in a vacuum. In the related field of information-seeking behaviour, several theoretical models have been proposed to describe information-seeking activities and propose hypotheses regarding factors that affect information needs of users. One of such models is the social model. Allen (1996, p. 74) noted that "*being a member of a group, such as abused spouses, cancer patients, senior citizens, or janitors, is seen as sufficient to influence individual information-seeking behaviors and patterns*". Indeed, most information-seeking research studies users as members of groups, often occupational groups (Case, 2002). Allen (1996) further noted that a user's information need is influenced by a complex set of cognitive factors specific to the user and factors shared among groups to which the user is a member. Several other information-seeking behaviour models include social, or more widely contextual, components in their attempts to describe users' information-seeking behaviour (Allen et al., 2011; Courtright, 2007). While such models of information-seeking behaviour do not reference the social role theory directly, it is clear that such models assume that some information needs originate in a context in which a user is affected by social factors, among others.

In my thesis, I build on the first three propositions of role theory that were mentioned earlier. I view each user as a member of several social positions. Each social position, an audience, is represented by a role, a set of behavioural insights that are characteristic of the position's members. In this work, the scope of behavioural insights is limited to those related to information-seeking and interaction. Specifically, I focus on content a user of a particular social position is expected to search for, consume, know, or need at some point as long as they remain a member of such a position. Such a representation could be words, a probability distribution over words, or even search tasks. While insights from other types of behaviour could be characteristic of certain social positions, this thesis focuses primarily on what information that members of a social position would interact with rather than, for example, how they interact with it.

This line of thinking is not new to the domain of user modelling. One of the earliest personalisation approaches relied on using individual characteristics of stereotypes to provide personalisation (Rich, 1979). A stereotype is a group of *similar* users. The similarity between users is determined using shared interests, tasks, background, or other personal features (Brusilovsky and Millán, 2007; Rich, 1979). Stereotypes can be used as a primary or supplementary source of information about each user (Shapira et al., 1997). In the primary mode, each user is assigned to one or several stereotypes and personalisation is performed using the stereotype's information (Brajnik et al., 1990; Chin, 1989). An inherent weakness in systems that apply stereotypes in the primary mode is the oversimplified assumption that all users belonging to a particular stereotype are identical to each other.

In contrast, stereotypes can be used as a supplementary source of information to initialise users' profiles when little information is known about them (Finin, 1989; Rich, 1979). Each user's profile is updated with individualised information as the system starts to collect information about the user. The collected information can enforce or contradict the stereotype information. For example, Rich's early work (1979) represented each user profile as a quadruple of attribute, value, rating, and justification. The rating element represents the system's confidence in the attribute's value. Rating is adjusted for each user during the user's interaction with the system. In such early systems, little attention was given to the challenge of defining and representing stereotypes. The predominant approach was based on qualitative analysis of users' responses to questionnaires (Shapira et al., 1997). However, such methods are unlikely to be suitable at web scale.

Alternatively, clustering methods represent a viable and effective approach in defining and representing stereotypes. After all, a stereotype is a cluster of similar users. Clustering is a feasible approach when rich information about users is available. Users can be grouped based on their location, social graphs, demographics, or their browsing behaviour. For example, Mei and Church (2008) placed users based on their IP addresses, which can be seen as an implicit indicator of a user's location, into five nested clusters. Their findings suggested that personalisation could be improved by augmenting a user profile with information from users in a nearby location when the current user's profile is sparse. Bennett et al. (2011) suggested estimating a probability distribution over users' locations for each web document using clickthrough data. They used the learnt probabilistic models as features in learning to rank settings and found that using such location-based features improves personalisation of search results. Personalisation using the geographical location appears to be effective for queries with geographical intent (Bennett et al., 2011) and seems to be applied by commercial search engines (Kliman-Silver et al., 2015).

In such work, similarity between users is calculated based on geographical proximity, which might implicitly reflect demographical similarity between users. Zhao et al. (2014) used demographical information explicitly in a product recommendation task while others (Bi et al., 2013; Hu et al., 2007; Jones et al., 2007) have shown that demographic traits can be inferred explicitly from web search behaviour. A different approach was followed by Carmel et al. (2009) who calculated the similarity between users based on their social graphs and re-ranked search results based on their relevance to people in the searcher's social graph. They experimented with three different methods to construct a user's graph: explicit or implicit familiarity indications, similar behaviour within the network such as co-usage of a tag or membership to similar communities, and a combined approach. They found that profiles built using social graphs improved personalisation over profiles that were constructed using the user's topical interests. Their work, however, was only evaluated for users within a single organisation.

Collaborative Filtering (CF) techniques provide an alternative mechanism for using similar users' data to complement a particular user's profile. While CF is a popular approach for recommendation systems such as news recommendation (Das et al., 2007), several researchers have applied it in the context of web search personalisation. Sugiyama et al. (2004) used a term vector to represent users' profiles based on their browsing history and CF methods were implemented to predict sparse terms' weights in an individual profile based on similar users' profiles. Xue et al. (2009) represented user profiles using three different language models. The first was an individual language model for each user based on the user's past queries and clicked documents. Then, profiles from all users were combined to form one global model. The third model is built by applying k-means clustering to users' profiles. The final user profile is the result of interpolating the user's profile, the user's cluster model, and the global model. According to Xue et al. (2009), this method provides a smoothing effect in scenarios like cold-start or for the issue of sparsity in individual users models. CF approaches have also been applied to predict missing clicks on certain URLs for a specific user based on logs of clickthrough data of all users (Sun et al., 2005).

Smyth (2007) referred to users with similar information needs as *a search community*. He argued that members of each community can benefit from the knowledge extracted from the community search behaviour as a whole. In a related piece of work, Freyne and Smyth (2006) described the design of I-SPY: a search engine intended to demonstrate the merits of incorporating information about search communities into traditional retrieval approaches. In their work, search communities can be defined based on the usage of topic-specific websites. For example, if a user submits a query using a search box located in a sport-related website then the user is more likely to be a member of a sport-related search community and seeking sports information. I-SPY used the clickthrough behaviour of the community members to calculate the relevance of a page to a query. They also analysed query logs from five different search applications and found queries submitted to vertical search applications[2] to show a degree of repetition. Search communities can also exhibit overlapping behaviour whereby the behaviour of a specific community can be used to complement another (Freyne and Smyth, 2006).

Teevan et al. (2009) formed groups based on several criteria such as topical interest, demographic, occupation, and search task. They examined the similarities of query selection, desktop information, and explicit relevance judgment among the members of each group in a controlled environment. Explicitly defined groups, e.g. those based on shared interests or tasks, were found to be similar with respect to group-related queries. They further showed that search results could be improved using group information for group-related queries.

[2] Image search, nutrition and fact findings search.

The previous research discussed above shares an overall assumption with the work presented in this thesis that search engine users can be grouped into clusters based on their behaviour. Group members would potentially benefit from the sharing of profile information among them. However, implicitly defining such groups without users explicitly declaring their membership to a particular group has proved to be a challenging task (Teevan et al., 2009). Much of the previous work has relied on clustering algorithms over users' interaction data to form user groups. Such methods presuppose the availability of rich interaction logs and user profiles, which is only available in proprietary settings. Furthermore, if user profiles are constructed using content-based methods, then the group models derived from such profiles only represent topical interests. A group model, as with a user model, needs not only to represent topical interests of the group members but also other important features such as search tasks and background as asserted in feature-based modelling (Brusilovsky and Millán, 2007). Taking inspiration from role theory, I present a novel approach to represent search engine users. This representation builds on two key assumptions. Firstly, the social position[3] of the user at query time may trigger the information need behind the search query. Secondly, each user is represented as a multinomial probability distribution over social positions.

The process I suggested in figure 3.1 aims to identify the intended audience for each document and to extract behavioural insights for each audience. An audience is equivalent in this thesis to a social position. Each social position is represented by a social role, which is the normative behavioural patterns, or insights, expected from a person occupying the position. In this process, the social position of a user is inferred based on the user's behaviour but the representations of social positions are learnt independently from any real user interaction. That is, a set of social positions and their representations are constructed from public web documents in a completely data-driven approach. A user with no previous interaction with the search engine will have a uniform distribution over this set of social positions. A social position with a high probability estimate in a particular user representation means that the user has frequently engaged in search behaviour, issuing queries or clicking on documents, believed to be relevant to the social position.

There are several benefits to this process relating to the three main challenges of building a framework to model users, as discussed at the beginning of this section. Firstly, the implicitly collected data about the user behaviour is used to map the user to the relevant social positions rather than as a representation on their own as done in most personalisation research. The web, as an open and rich resource, is the main source[4] of information used to build the representation of social positions. Secondly, this process has

[3] For example, programmer or movie fan.

[4] In theory, users' interaction data can be used to update the representation of social positions in parallel with public web documents. In this thesis, however, I exclusively relied on public web documents to extract and represent social positions.

the potential to lessen privacy issues that might hinder personalisation research. Social positions and their representations are public information derived from public data. The openness of such a representation allows entities other than search engines to build and maintain social positions and roles. For example, the user can download a bundle of social position representations to enhance his/her local profile or to re-rank search results according to certain social positions in client-side privacy-aware settings. Thirdly, social position representations are extracted from web documents that are more structured and linguistically governed compared with the unstructured and idiosyncratic nature of search queries (Bendersky et al., 2011b).

The structure of the information source enables the application of NLP algorithms that rely on grammatical features of sentences. Therefore, it would be feasible to represent a social position by a probability distribution over tasks, or events, extracted from documents believed to be relevant to the social position. This procedure would enable feature-based modelling in which users are represented via their interests and tasks. For example, a web document relevant to the social position of *a professor* might contain terms such as *university, students, exams, research, conference, publication,* and *committee.* It is also likely to contain phrases referring to tasks such as *submit a grant proposal, review research papers* or *examine a student.* In general, this approach provides a principled and computationally feasible framework to reason and generalise about user behaviour. It should be noted that people can recognise social positions and social roles (Biddle, 2013; Wasserman and Faust, 2009). This societal fact has a potentially favourable implication on the proposed framework. It might indicate that search engine users would easily understand social positions as labels of their user models.

At this point, it is important to note several areas of ethical concerns regarding the approaches taken in my thesis. First, I rely on public web documents to learn a representation of social positions. As a large and rich resource for various applications, the web is also a source of offensive, biased, and racist content. Such a problem is already recognised in the NLP domain (Bender et al., 2021; Bolukbasi et al., 2016; Schick et al., 2021; Sheng et al., 2019). A social position's representation that is induced from web documents would inherit the source's biases unless measures are taken to lessen such an effect. While this is still an active area of research, several possible solutions have been proposed in the literature. For example, it might be feasible to measure the sentiment of the learned representations and exclude negative ones (Sheng et al., 2019). However, it would still be needed to define negative examples in the context of social roles. Another possible approach is to define a modelling component that can diagnose and reduce biases in representations based on minimal supervision (Schick et al., 2021). The second source of concern is that specific types of social positions that are based on societal, religious, ethnic, sexual, and physical characteristics are sensitive by nature. Users might not want to reveal

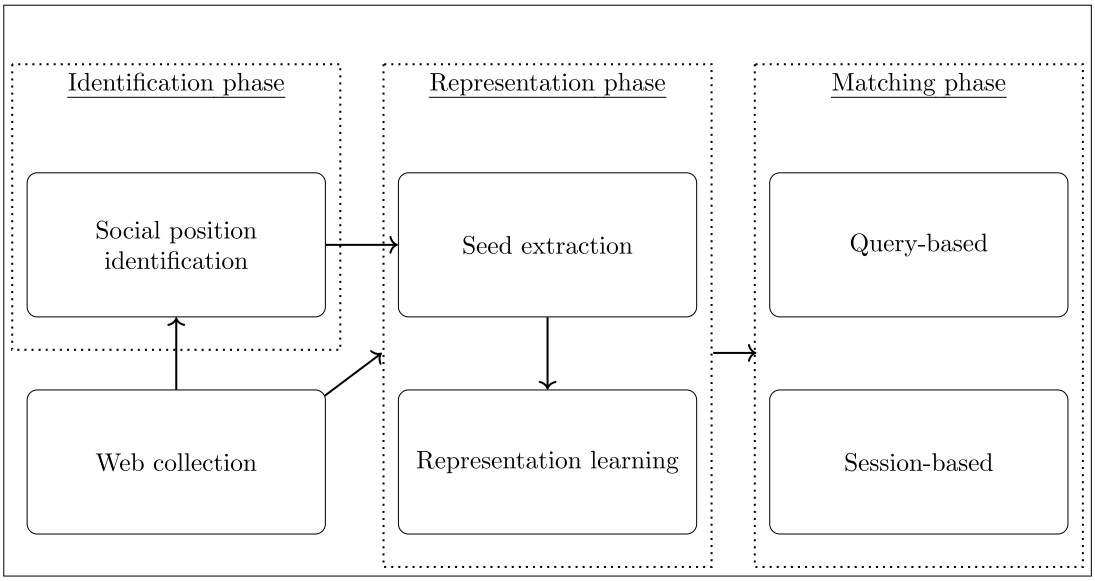


Figure 3.2: An overview of the framework components.

their membership to certain social positions. They might also disagree with stereotypical characteristics that might be learned to represent such social positions. These are open research areas for future investigation.

There are three main components of the proposed framework. The first concentrates on identifying social positions; as discussed earlier, a social position is a label for related behavioural insights. The second component involves building a representation for each social position. Both of these components use web documents to identify and represent social positions. The third component focuses on matching a query or a search session to its most relevant social positions. These three components are illustrated in figure 3.2. This chapter focuses on the first component. In section 3.2, I formulate the identification of social positions as a binary classification task. Section 3.2.3 details the experimental settings. The proposed approach is validated in section 3.3 while section 3.4 presents a summary of this chapter.

3.2 Social position identification

3.2.1 Task formulation

Operationally, a social position is a label for a set of related behavioural patterns. It is often the case that such positions are expressed in simple linguistic labels (Wasserman and Faust, 2009) using nouns or noun phrases. If users were to be asked to associate themselves or others to a community or an audience, the answers would typically follow simple linguistic patterns. Examples of such patterns are underlined in "*I am a* computer scientist", "*he was a* Linux user" and "*she is an* IR researcher". Intuitively, applying such patterns to naturally occurring sentences would enumerate a large set of social positions. However, the quality of the extracted candidate positions depends on the quality of the data source from which sentences are extracted. In this thesis, the data source is a large corpus of general web documents. Whilst a web corpus can be considered representative of the target user population, the quality of the extracted labels is likely to be limited. Vague labels such as *fan*, *user,* and *member* are commonly used in the web which is challenging because each label has to unambiguously point to a community of users. I use a set of lexical patterns in the identification task as a fast and reasonable initial approach for the enumeration of candidate nouns and noun phrases. These lexical patterns are:

- I am [a/an] NP.
- She [is/was] [a/an] NP.
- He [is/was] [a/an] NP.

I, therefore, formulate social position identification as a binary classification task. The input to the classifier is the set of candidate positions extracted using the lexical patterns. The output is a binary judgment about whether a candidate noun or noun phrase is a social position or not. The key criteria to classifying a candidate as a social position is that each candidate label must have one predominant semantic interpretation that is not context-dependent. For example, *traveller*, *computer science student*, *football fan* and *computational linguist* have one dominant semantic interpretation. This does not mean that each social position must have only one semantic interpretation but that there is at least one dominant semantic interpretation that refers to a person and can be associated with at least one topic[5]. Eventually, each social position is a label for a community of users with similar interests and if the label has at least one dominant sense that refers to a person then a coherent representation for the interests of the social position can be built algorithmically. On the other hand, labels such as *fan*, *user*, *enthusiast* or *member* would have varying interpretations depending on the context in which they are used and thus do not necessarily refer to a cohesive community with shared interests.

A social position is either a noun or a noun phrase. For annotation purposes, each noun phrase is divided into two parts: a head noun and a concept segment[6] which can take any syntactic form itself. The head noun is always the right-most noun. For example, the head noun for the social position "breast cancer patient" is *patient* while *breast cancer* is the concept segment. The rationale behind this is that social positions in the form of noun phrases often point to an area or a topic from a specific viewpoint. For example, social positions such as *breast cancer survivor*, *breast cancer activist* and *breast cancer oncologist* refer to the topic of *breast cancer* from the viewpoints of *survivor*, *activist* and

[5] Topic as in a topic-based taxonomy or a coherent probability distribution of related terms.

[6] The dependent part of a noun phrase.

Feature Set	Feature description
FS1	(1) Number of times it is preceded by the lexical patterns.
	(2) Number of times it is preceded by an adjective.
	(3) Number of times it is preceded by a determiner.
	(4) Count of ClueWeb09 occurrences.
FS2	(5) Ends with *er*.
	(6) Ends with *ist*.
	(7) Contains *and*.
	(8) Contains special characters.
FS3	(9) Followed by a verb phrase.
	(10) Preceded by a verb phrase.
FS4	(11) Followed by a noun phrase.
	(12) Preceded by a noun phrase.
FS5	(13) Followed by a preposition.
	(14) Preceded by a preposition.
FS6	(15) Tagged as a person by the C&C package.
	(16) Preceded by the pattern NP is|was such that NP is a name of a person.
FS7	The preceded adjectives.
FS8	Verbs preceding or following the candidate position.

Table 3.1: Features used to train the social position identification classifier. Features in the sets 3, 4, 5 and 6 represent count of occurrences. All features were normalised linearly by the value of feature 4. Feature 4 is normalised by its min/max values.

oncologist respectively. Therefore, it is necessary to establish that the concept segment refers to a topic and the head noun is a type of person. Social positions made up from a single noun will not have a concept segment and that is acceptable. Table 3.1 lists the different sets of features that are used for the classification experiments.

3.2.2 Classification algorithm

I use Adaptive Regularisation of Weight Vectors (AROW) as the classification algorithm (Crammer et al., 2009). AROW is an example of an online learning algorithm that operates in rounds and processes one training instance at a time. At time step t during training, AROW applies its current model $\boldsymbol{w}$ on instance $\boldsymbol{x}_t$ to produce its predicted label $\hat{y}_t$ such that $\hat{y} = h_{\boldsymbol{w}}(\boldsymbol{x}) = sign(\boldsymbol{w} \cdot \boldsymbol{x})$ for binary classification which is the case for social position identification. Then, AROW would receive the true label y_t for instance $\boldsymbol{x}_t$ and suffer a loss $\ell(y_t, \hat{y}_t)$. The loss is 0 for correctly classified instances and 1 otherwise. The model $\boldsymbol{w}$, or the weight vector, is drawn from the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. In practice, $\boldsymbol{w}$ often corresponds to the average weight vector of the mean $\boldsymbol{\mu} \in \mathbb{R}^d$. $\Sigma \in \mathbb{R}^{d \times d}$ is a covariance matrix. The value $\Sigma_{i,i}$ represents the model's confidence in feature i.

AROW is updated according to the following equation at each round:

$$\mathcal{C}(\boldsymbol{\mu}, \Sigma) = D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})) + \lambda_1 \ell_{h^2}(y_t, \boldsymbol{\mu} \cdot \boldsymbol{x}_t) + \lambda_2 \boldsymbol{x}_t^\top \Sigma \boldsymbol{x}_t \tag{3.1}$$

where,

$$\begin{aligned} \ell_{h^2}(y_t, \boldsymbol{\mu} \cdot \boldsymbol{x}_t) &= (max\{0, 1 - y_t(\boldsymbol{\mu} \cdot \boldsymbol{x}_t)\})^2 \\ \lambda_1 = \lambda_2 &= \frac{1}{2r} \quad \text{for} \quad r > 0 \end{aligned}$$

$$\begin{aligned} (\boldsymbol{\mu}_t, \Sigma_t) = \quad & \min_{\boldsymbol{\mu}, \Sigma} D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})) \\ & \text{s.t.} \quad P_{\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)}[y_t(\boldsymbol{w} \cdot \boldsymbol{x}_t) \geq 0] \geq \eta \end{aligned} \tag{3.2}$$

The first term in equation 3.1 conveys the attempt to preserve as much information learnt in the previous training rounds as possible. Formally, this is expressed using the KL-divergence between the new and old distributions and is used in other online learning algorithms such as Confidence Weighted (CW) learning (Dredze et al., 2008) in equation 3.2. AROW makes its novel extension in the second and third parts of the equation. The second component of equation 3.1 means that the new model should predict $\boldsymbol{x}_t$ with low loss. This softens the hard constraint in CW that requires the prediction of the correct label for the current instance $\boldsymbol{x}_t$ to be made with a probability $> \eta$ where η must be > 0.50. The third term asserts that the model confidence should grow as a result of processing more training instances. These novel modifications allow AROW to perform particularly well in the presence of noisy labels and avoid over-fitting. Algorithm 2 presents the pseudocode for AROW (Crammer et al., 2009).

3.2.3 Settings

The experiment in this section uses the ClueWeb09 category B dataset (Callan et al., 2009), which consists of about 50 million English web documents that were crawled in 2009. I extracted sentences from each web document using the OpenNLP library[7] and sentences were POS-tagged and chunked using the C&C package (Clark and Curran, 2007). Approximately 743 million sentences were extracted in total[8]. Using the lexical patterns discussed in section 3.2, I found $915,407$ sentences that contained candidate social positions. Since this is a supervised classification task, I annotated a training dataset which consisted of 2000 training instances divided into 1000 positive social positions and another 1000 negative samples. As discussed earlier, each candidate position must have at least one dominant interpretation that is not context-dependent. This requirement is

[7] https://opennlp.apache.org/

[8] Sentences that were not processed by the C&C package were removed.

Algorithm 2: AROW classification algorithm (Crammer et al., 2009).

```
Input: Parameter $r$
Output: Weight vector $\boldsymbol{\mu}_T$ and confidence $\Sigma_T$
begin
    Initialization step:
    $\boldsymbol{\mu}_0 = \mathbf{0}, \Sigma_0 = I$
    foreach $t = 1...T$ do
        Receive a training example $\boldsymbol{x}_t \in \mathbb{R}^d$ ;
        Compute margin and confidence $m_t = \boldsymbol{\mu}_{t-1} \cdot \boldsymbol{x}_t \quad v_t = \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t$ ;
        Receive true label $y_t$, and suffer loss $\ell_t = 1$ if $sign(m_t) \neq y_t$ ;
        if $m_t y_t < 1$, then
            $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + \alpha_t \Sigma_{t-1} y_t \boldsymbol{x}_t \qquad \Sigma_t = \Sigma_{t-1} - \beta_t \Sigma_{t-1} \boldsymbol{x}_t \boldsymbol{x}_t^\top \Sigma_{t-1}$;
            $\beta_t = \frac{1}{\boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t + r} \qquad \alpha_t = max(0, 1 - y_t \boldsymbol{x}_t^\top \boldsymbol{\mu}_{t-1}) \beta_t$;
        end
    end
end
```

operationalised through the association of each candidate position to a topic. If at least one topic was frequently associated with the position, then it would be judged as a positive social position. I used Wikipedia pages of the social position and the concept segment, if any, to implement this requirement. For example, the social position of *tourist* can be represented using the topic of *tourism* or *travel* while the position of *professor* would be represented by topics such as *academia* or *university.* I also consider any social position that is based on personality traits such as *smart*, *patient,* or *confident* to be negative examples of social positions. This thesis focuses exclusively on social positions that can be identified and represented using content-based methods only. Training and testing are performed in a five-fold cross validation setting and I report the average score in terms of classification accuracy. Statistical tests were computed using the pairwise Wilcoxon rank-sum test at 0.05 significance level. I set the value of r in the AROW algorithm to 0.1, which is the default value and the number of rounds to 1000. To investigate the reliability of my annotation process, I have asked three participants to re-annotate the complete dataset (2000 social positions). All participants were native English language speakers and were provided with the same annotation guidelines. To measure the inter-annotator agreement, I used Cohen's kappa coefficient as defined in the following formula (Cohen, 1960):

$$\kappa = \frac{P_o - P_c}{1 - P_c} \tag{3.3}$$

where P_o is the observed agreement between annotators and P_c is the proportion of annotators' agreement that is expected by chance. The kappa value can range from perfect disagreement (-1) to complete agreement $(+1)$. A kappa value of 0 indicates that agreement is expected only by chance. There are several scales that can be used to

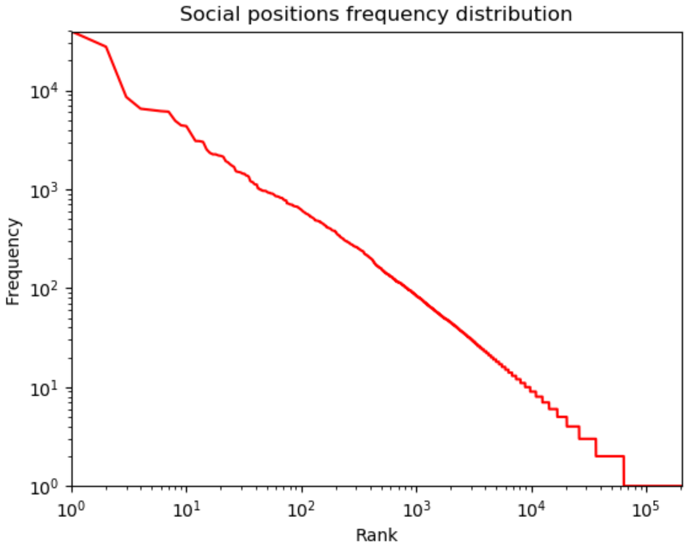


Figure 3.3: Candidates frequency distribution in a log-log scale.

interpret the kappa value. For example, Fleiss et al. (2003) consider $\kappa > 0.75$ to indicate excellent agreement beyond chance while Krippendorff (1980) require $\kappa \geq 0.80$ to show reliable agreement. According to Krippendorff, a kappa value in the range $0.67 < \kappa < 0.80$ allows a tentative conclusion to be drawn. For the social positions annotation task, I found $\kappa = 0.77$ $(N = 2000, k = 4, n = 2)$[9]. This value might indicate reasonable above chance agreement between the annotators on identifying candidate social positions. Since the provided guidelines were quite straightforward, this kappa value might suggest that participants tended to agree on what constitutes a valid social position.

3.3 Results

Using the lexical patterns described in section 3.2, I extracted $204,781$ distinct candidate social positions. The frequency distribution of such candidate positions is presented in figure 3.3. It shows that few candidate positions are frequently used in combination with the lexical patterns while many others are rarely seen with such lexical patterns[10]. Table 3.2 shows the top 20 candidate social positions and the frequency of their occurrences as the noun or noun phrase component in the lexical patterns. These 20 candidates account for about 15% of the total sum of occurrences while 68% of the candidates are encountered only once. These empirical findings suggest that the use of lexical patterns may not be

[9] I have included my original annotation as the fourth annotator.

[10] I applied the statistical method of Clauset et al. (2009) to test if this dataset can be described by the power law distribution. The result indicated that the power law is not a plausible hypothesis ($p = 0.04$). I used the R package developed by Gillespie (2014).

Candidate position	**Frequency**	**Candidate position**	**Frequency**
member	39454	fellow	3627
fan	27555	huge fan	3078
graduate	8562	adult	3065
man	6516	professor	3007
little	6319	mother	2528
bit	6160	artist	2336
big fan	6081	christian	2254
student	4923	writer	2254
expert	4425	person	2195
woman	4346	friend	2173

Table 3.2: Top 20 extracted social positions frequencies.

ID	**Features**	**Accuracy (%)**
1	FS1 (baseline)	72.15
2	FS1 + F2	74.10
3	FS1 + FS2 + FS3	74.75
4	FS1 + FS2 + FS3 + FS4 ●	79.40
5	FS1 + FS2 + FS3 + FS4 + FS5 ●	79.85
6	FS1 + FS2 + FS3 + FS4 + FS5 + FS6 ●	79.55
7	FS1 + FS4 ●	76.70
8	FS1 + FS2 + FS4 ●	78.65
9	FS7 ●↑	82.50
10	FS8 ●	80.45
11	FS7 + FS8 ●↑	83.80
12	All features ●↑	**85.80**

● Indicates a statistically significant improvement over the baseline.
↑ Indicates statistically significant improvement over classifier 6.

Table 3.3: Social position identification results. Statistical significance is calculated using the pairwise Wilcoxon rank-sum test.

a sufficient method in identifying social positions but rather an enumeration method to provide a set of candidates for further classification. For instance, ambiguous candidates such as *member*, *fan*, *little,* and *bit* are commonly used in association with the lexical pattern while valid social positions such as *slow dancer*, *crop artist*, *army medic,* and *django developer* are seen only once.

Table 3.3 presents the results of 12 configurations of the classification features. As previously displayed in table 3.1, I developed eight sets of features. The first (FS1) includes the number of times a candidate position is preceded by a lexical enumeration pattern, an adjective, and a determiner. It also includes the candidate's number of occurrences in ClueWeb09 dataset. The second set (FS2) indicates if a candidate social position ends with *er*, *ist* or contains an *and* or a special character. FS3, FS4 and FS5 collect statistics for the number of times a candidate is followed or preceded by a verb phrase, a noun phrase or a preposition, respectively. The sixth feature set (FS6) includes two features: the number of times a candidate is tagged as a person by the C&C package and the frequency of the candidate being preceded by the pattern *NP is*|*was* such that *NP* is a name of a person. FS7 includes the preceding adjectives while FS8 are verbs that precede or follow a candidate social position. In the following, I refer to each configuration by its number. As shown in the table, the first set of features scores reasonably even though it consists of four basic features (accuracy score of 72%). To explore the performance of FS1 features, I investigated a few permutations of the four features and found that most of the gain in performance seemed to be attributed to feature No. 2. That is, by removing feature No. 2 from FS1, the accuracy dropped to around 50% which might suggest that candidate social positions frequently preceded by an adjective are more likely to be labelled as positive instances.

Upon analysis, this classifier appears to enforce, albeit roughly, the soft constraint that deems candidate social positions that are modified by an adjective as negative instances. This results in several false negative and false positive cases such as *medical doctor* or *candidate*, respectively. I treat this classifier as the baseline in my experiments. Classifier 2 added another four quality features to take advantages of frequent morphological patterns associated with social positions (features No. 5 and 6) and to implement the second hard constraint that candidate social positions should not be composed of multiple positions (feature No. 6). This resulted in a modest, although not statistically significant, improvement from baseline. One of the hypotheses I studied is the semantic role of the candidate position being a patient or an agent of a verb phrase (classifier 3). The mere frequency-based features did not seem to provide noticeable improvement over classifier 2 but the inclusion of the preceded or succeeded verbs as the only features, as in classifier 10, seemed to achieve a positive result (accuracy = 80.45%) that was statistically significant compared to the baseline and classifier 3.

One possible interpretation of the performance of classifier 10 is that a social position is essentially a named-entity of type human and would have a distribution over verbs that might selectively prefer to have a human named-entity as their subjects or objects, a linguistic concept known as selectional preference (Clark and Weir, 2002; Resnik, 1993; Séaghdha, 2010). The performance of classifier 9, which used the preceded adjectives as the only features, might also be the result of a similar behaviour to that of verbs-based classifier (accuracy = 82.50%). That is, a set of adjectives might often modify human named-entities. For example, both classifiers (9 and 10) correctly labelled instances such as *fish*, *bird*, *creature* and *beast* as negative instances in contrast to classifiers 1, 2 and 3. Table 3.4 lists the top 15 features for classifiers 9 and 10.

In classifier 4, I studied the effect of including features that capture the location of candidates within matched noun phrases. More specifically, I included a feature to indicate

Adjective	Weight	Verb	Weight
part-time	73.87	P-appoint	68.68
future	68.43	F-design	66.52
lifelong	62.73	F-travel	65.37
prospective	62.55	F-live	61.03
contemporary	60.50	F-create	58.77
traditional	54.94	F-want	58.33
tenured	48.52	F-visit	53.95
passionate	48.26	P-improve	51.07
adopted	48.04	P-confirm	50.72
influential	46.94	P-hold	50.25
avid	46.82	F-programme	49.68
observant	45.56	P-speak	49.56
northern	45.15	F-research	48.90
foster	44.86	F-order	48.46
ardent	43.86	P-recover	47.11

Table 3.4: Top 15 adjective and verb features. P and F refer to preceded and followed.

how often the candidate position starts a noun phrase (feature No. 11) and how often it does not (feature No. 12). Their inclusion seemed to provide statistically significant improvement (accuracy = 79.40%) over baseline. By using FS1 and FS4 as in classifier 7, the performance was significantly increased compared with FS1 only which suggests that the location of candidate positions within noun phrases played a positive role in improving the classification accuracy. I analysed this behaviour further by removing feature No. 12 from classifier 7, which resulted in an accuracy score of 76.05% compared with 73.90% when removing feature No. 11 from the same classifier. This might indicate that the improvement in classification accuracy is largely due to feature No. 11.

To examine this result further, I ranked the candidate positions based on the relative frequency of their occurrences at the beginning of a noun phrase normalised by their total frequency of occurrences (see table 3.5). Social positions based on ethnicity, political, and religious affiliations dominated the top results, which could indicate that classifier 4 performed reasonably well at labelling such positions. For example, positions such as *European*, *Canadian*, *liberal*, *republican*, *Christian,* and *Muslim* were all classified incorrectly as negative instances by classifiers 1,2 and 3 but correctly as positive by classifier 4. Overall, the best performance was achieved when all features were included (accuracy = 85.80%) although this result was not statistically significant compared with using adjectives and verbs (classifier 11) as the only features.

Analysis of the inter-annotation agreement indicated a reasonable above chance agreement on labelling social positions ($\kappa = 0.77$). However, this task might still be affected by subjective judgment. For example, two of the annotators labelled the positions *employer* and *student* as negative because it appeared to them as broad and not specific while the

Rank	Candidate position	Score	Rank	Candidate position	Score
1	european	0.9785	11	brazilian	0.9600
2	israeli	0.9717	12	humanitarian	0.9598
3	colombian	0.9685	13	regular	0.9588
4	canadian	0.9670	14	academic	0.9579
5	african	0.9668	15	south african	0.9568
6	australian	0.9661	16	nigerian	0.9536
7	iranian	0.9660	17	ethiopian	0.9524
8	iraqi	0.9644	18	mexican	0.9459
9	palestinian	0.9639	19	pakistani	0.9447
10	postgraduate	0.9618	20	scandinavian	0.9436

Table 3.5: Example social positions.

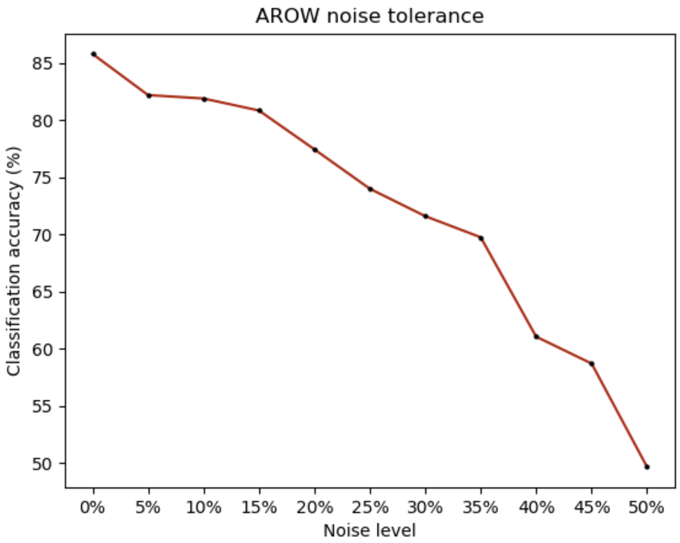


Figure 3.4: AROW noise tolerance.

remaining annotator labelled it as positive[11]. The subjectivity of annotation can be seen as a source of noise in the training set. To investigate this I created noise by randomly selecting training instances uniformly and changing their labels from positive to negative and vice versa. Figure 3.4 shows the effect noisy training data had on classification accuracy. AROW maintained a reasonable classification accuracy above a score of 80% in the presence of a modest amount of noise (from 5 to 15%). However, the accuracy started to degrade sharply when the noise level was above 35%. In general, this result might suggest that social positions can be identified with a reasonably high accuracy even in the presence of a moderate amount of noise in the labelled dataset.

[11] They are labelled as positive in the original annotation.

3.4 Summary

This chapter introduced a novel framework to model search engine users. In section 3.1, I presented the framework's general characteristics and components and also contained background discussion to related group-based modelling approaches. The proposed framework consisted of three main phases: social position identification, representation, and matching. The first component was detailed in this chapter. In section 3.2, I formulated the task of identifying social positions as a binary classification task and I developed a method to enumerate candidate social positions from web documents. These candidates formed the input to a classification algorithm based on the AROW algorithm as explained in section 3.2.2. The experimental settings were discussed in section 3.2.3. The results, as in section 3.3, showed that social positions can be identified with reasonably high accuracy using a set of novel features. In the next chapter, I present a method to build a representation for each social position.

CHAPTER 4

Learning social positions representation

This chapter is concerned with the learning of user models for social positions. The process I presented in figure 3.1 aims at identifying the intended audience, i.e. social position, for each web document and then extracting behavioural insights, i.e. user model, to be used in representing the identified audience. The first phase in implementing such a process is the identification of social positions as in chapter 3. The identified social positions are essentially concise labels in the form of nouns or noun phrases. In this chapter, I present a method to learn a rich semantic representation for each social position. Such a representation is needed to identify documents of interests to each social position from which additional representations can be learned. In other words, a user model for each social position must be learnt.

4.1 Background

In any user modelling task, there are three key questions to be answered (Brusilovsky and Millán, 2007). These are: what aspect of the user is being modelled, how is it going to be represented, and how would the model be maintained. There is also a dependence on the method by which users' information is gathered. As discussed previously, most personalisation techniques use implicit approaches to model users based on the content they interact with. This data collection method mostly captures users' interests and some factors of their context (e.g. location and time) but not, at least directly, users' knowledge, tasks, or personality traits. It also influences the structure of the user model itself to be one of those used to represent content such as keywords vectors (e.g. Matthijs and Radlinski, 2011; Sugiyama et al., 2004), concept networks (Micarelli and Sciarrone, 2004), topic taxonomies (e.g. Bennett et al., 2012; Sontag et al., 2012), or topic modelling (e.g. Harvey et al., 2013; Vu et al., 2015).

In this thesis, models for social positions are also content-based. They primarily represent the interest aspect as most personalisation approaches do. The difference, however, is that our approach relies on web documents to build a model for each social position independently from users' interactions. This process allows for distributed, transparent, and continuous maintenance for each social position's model. Further insights and aspects about each model such as search tasks could be learnt from web documents that are relevant to the social position and used to update the respective model accordingly. The structure of social positions' models is based on the concept of *topics* from the topic modelling literature. In the following section, a review of existing topic modelling approaches is presented. Section 4.2 formulates the task of representing social positions as a topic modelling task. Sections 4.3 and 4.4 discuss the experimental settings and results, accordingly. Lastly, this chapter is summarised in section 4.5.

4.1.1 Topic modelling

Words that make sense together form a topic (Boyd-Graber et al., 2017). The mathematical modelling of such topics mostly builds on distributional semantics' concepts (Clark, 2015; Turney and Pantel, 2010); mainly the hypothesis that words that tend to occur in similar contexts have similar meanings (Firth, 1957; Harris, 1954). The need for such modelling can be motivated by two major uses. Firstly, textual content has and continues to grow at an increasing rate which calls for methods that can represent such a large quantity of text in compact representations to enable discovery and explorations. Secondly, in IR applications, the issue of mismatch between a query's terms and terms used in a relevant document has posed a major challenge in the field, which is known as the vocabulary mismatch problem. Classical approaches such as stemming and query expansion have been proposed to tackle such an issue in addition to topic modelling approaches (Croft et al., 2010). In this section, I describe two prominent topic modelling approaches: Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA). The latter is used to model social positions.

4.1.1.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA)[1] was introduced as a candidate solution for two pressing problems in term matching retrieval (Deerwester et al., 1990). The first was the vocabulary mismatch problem or synonymy where a relevant document might not contain the same words that the user used to express their information need although both might be on the same topic. The second issue is polysemy where some words have multiple interpretations and the occurrence of a query's word in a document does not necessarily mean they both refer to the same sense. The assumption behind LSA is that a latent, higher-order, and compact structure might be extracted from a document collection. Such a structure could represent associations between words and documents. In such settings, the matching

[1] Also known as Latent Semantic Indexing.

of a query to a document is based on projecting both into a latent semantic space and comparing the similarity between their latent vector representations.

Similar to the Vector Space Model (VSM), LSA starts with an $N \times M$ matrix, henceforth X. N is the number of terms in the document collection while M is the number of documents. X is a term-document matrix where each element e_{ij} in X represents the frequency, or weighted score, of term i in document j. LSA uses Singular Value Decomposition (SVD) to represent X using the product of three matrices:

$$X_{nm} = U_{nn} S_{nm} V_{mm}^T \tag{4.1}$$

where U and V are orthonormal and S is diagonal and contains the singular values in descending order along its diagonal. To perform dimensionality reduction, S gets restricted to the first K values such that:

$$X \approx U_{nk} S_{kk} V_{km}^T \tag{4.2}$$

Similarity between words can be computed from the matrix US and between documents from VS.

Rather than representing documents and words as points in Euclidean space, a statistical family of models defines a topic as a probability distribution over words (Steyvers and Griffiths, 2007). These models are generative in nature and still rely on the co-occurrence hypothesis. The higher order representation that such models seek to infer is supposed to describe how words in each document might have been generated. Each document is a mixture of topics and each topic is a multinomial distribution over words. Note that such models do not attempt to explain the order of words occurrences, i.e. these are bag-of-words models. A prominent example of such is the Probabilistic Latent Semantic Analysis (pLSA) model (Hofmann, 1999). pLSA attempts to associate an unobserved latent class variable z with each word in the document collection. The number of classes, or topics, K is set a priori. Figure 4.1 shows pLSA in the plate notation. The generative story of pLSA is as follows:

- Select a document d with probability $P(\theta_d)$.

  1. Pick a topic z with probability $P(z|\theta_d)$.
  2. Generate a word w with probability $P(w|z)$.

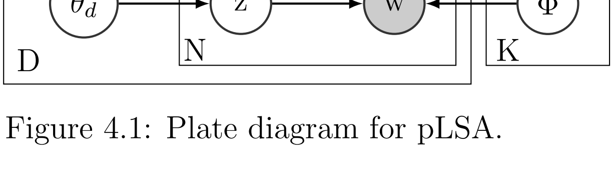


Figure 4.1: Plate diagram for pLSA.

The joint probability of this model translates to:

$$P(\theta_d, w) = P(\theta_d) \sum_{z \in Z} P(w|z) P(z|\theta_d) \tag{4.3}$$

where Z is the set of all topics. One of the major issues with pLSA is that it does not make any assumption about how the mixture weight θ_d is generated which makes it difficult to generalise to unseen documents (Steyvers and Griffiths, 2007).

4.1.1.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) overcomes pLSA's limitations by introducing a Dirichlet prior to the document's topic mixture variable θ_d and the topics' variable Φ. Both variables θ_d and Φ are multinomial and the Dirichlet distribution is a conjugate prior for the multinomial distribution. This simplifies the issue of statistical inference and enables LDA to generalise over unseen documents. Again, each document is modelled as a mixture of topics and each topic is a probability distribution over words. The sound probabilistic basis and relative ease of inference and generalisation have made LDA the main algorithm for topic modelling in various applications, including IR (Wei and Croft, 2006). It also forms the building block of many specialised models in areas such as authorship profiling (Rosen-Zvi et al., 2004), sentiment analysis (Lin and He, 2009), and hierarchical topic modelling (Griffiths et al., 2003). Figure 4.2 presents the plate diagram of LDA while the generative story is as follows:

- For each document d in D:
  - Choose $\theta_d \sim Dir(\alpha)$.
  - For each word w_n in document d.
    1. Choose a topic assignment $z_n \sim Multi(\theta_d)$.
    2. Choose a word $w \sim Multi(\Phi^{z_n})$.

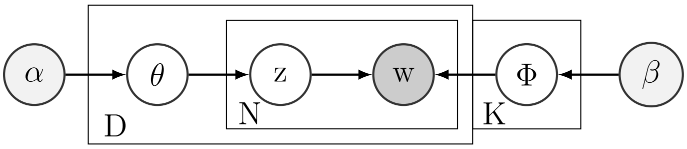


Figure 4.2: Plate diagram for LDA.

The posterior probability of such a model can be inferred using methods such as variational inference (Blei et al., 2003) or Gibbs sampling (Griffiths and Steyvers, 2004). In this thesis, I use the collapsed Gibbs sampling method for its simplicity and efficiency (Boyd-Graber et al., 2017; Griffiths and Steyvers, 2004). Gibbs sampling is a Markov

Chain Monte Carlo (MCMC) method used to estimate joint probability distributions given a set of samples. It iteratively updates a sampled latent variable given the current values of the other variables. In LDA, the full collection joint probability can be written as follows:

$$P(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\Phi}|\alpha, \beta) = \underbrace{P(\boldsymbol{\theta}|\alpha)}_{\text{Dirichlet}} \underbrace{P(\boldsymbol{z}|\boldsymbol{\theta})}_{\textit{Categorical}} \underbrace{P(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\Phi})}_{\textit{Categorical}} \underbrace{P(\boldsymbol{\Phi}|\beta)}_{\text{Dirichlet}} \tag{4.4}$$

The collapsed Gibbs sampling works firstly by integrating out the θ and Φ which results in the following joint probability:

$$P(\boldsymbol{w}, \boldsymbol{z}) = P(\boldsymbol{z})P(\boldsymbol{w}|\boldsymbol{z}) \tag{4.5}$$

Now, suppose we have two count matrices as the following:

1. A matrix of dimension $D \times K$ called Ω, where D is the number of documents and K is the number of topics. Ω_d refers to the d^{th} row in Ω. $\Omega_{d,k}$ refers to how many times tokens from topic k are found in document d.

2. A matrix of dimension $V \times K$ called Ψ, where V is the numbers of entries in the vocabulary. Ψ_v refers to the v^{th} row in Ψ. $\Psi_{v,k}$ refers to how many times word v is assigned to topic k.

Each word in the document collection will be assigned a random topic label z. The number of topics K is set a priori. We can re-write the two components of equation 4.5 as follows:

$$P(\boldsymbol{z}) = \prod_{d=1}^{D} \frac{\Gamma\big(\sum_{k=1}^{K} \alpha_k\big)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(\Omega_{d,k} + \alpha_k)}{\Gamma\big(\sum_{k=1}^{K} \Omega_{d,k} + \alpha_k\big)} \tag{4.6}$$

$$P(\boldsymbol{w}|\boldsymbol{z}) = \prod_{k=1}^{K} \frac{\Gamma\big(\sum_{v=1}^{V} \beta_v\big)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \frac{\prod_{v=1}^{V} \Gamma(\Psi_{k,v} + \beta_v)}{\Gamma\big(\sum_{v=1}^{V} \Psi_{v,k} + \beta_v\big)} \tag{4.7}$$

By dropping out constant terms, we get the following equation:

$$P(\boldsymbol{w}, \boldsymbol{z}) \propto \left(\prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(\Omega_{d,k} + \alpha_k)}{\Gamma\big(\sum_{k=1}^{K} \Omega_{d,k} + \alpha_k\big)} \right) \times \left(\prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(\Psi_{v,k} + \beta_v)}{\Gamma\big(\sum_{v=1}^{V} \Psi_{v,k} + \beta_v\big)} \right) \tag{4.8}$$

Then, the sampler iterates over each word and estimates $P(z_i = j|\boldsymbol{z}^{\neg i}, \boldsymbol{w})$ which is the probability of assigning it to each topic given all the other word topic assignments except the assignment of the current token. $P(z_i = j|\boldsymbol{z}^{\neg i}, \boldsymbol{w})$ can be estimated using the following equation (Griffiths and Steyvers, 2004)[2]:

$$P(z_i = j|\boldsymbol{z^{\neg i}}, \boldsymbol{w}) \propto \frac{\Omega^{\neg i}_{d_i,j} + \alpha}{\Omega^{\neg i}_{d_i,*} + K\alpha} \times \frac{\Psi^{\neg i}_{w_i,j} + \beta}{\Psi^{\neg i}_{*,j} + V\beta} \tag{4.9}$$

where:

1. $\Psi^{\neg i}_{v,k}$ is the number of times term v is assigned to topic k with the assignment of the i^{th} word excluded.

2. $\Omega^{\neg i}_{d,k}$ is the number of times topic k is assigned to document d with the assignment of the i^{th} word excluded.

3. d_i is the current document and w_i is the current word to be assigned a topic.

In this thesis, I use an LDA-based method to represent each social position model as a multinomial distribution over words, i.e. a topic. The rationale behind such an approach is fourfold. Firstly, it is realistic to assume that a particular set of words is more commonly used in the context of a specific social position than others. For example, a tourist may use words like *travel, airline, hotel, attraction,* and *holiday* more frequently than a programmer. Secondly, each position is a distribution over all words as some positions may have overlapping common words. Thirdly, the probabilistic foundation of such an approach enables efficient maintenance of the models. Each social position's model can be updated when more documents are found to be relevant to the social position. Lastly, users can be represented as a mixture of social positions.

4.2 Task formulation

I have developed a topic modelling-based approach to represent social positions which consists of two main components. The first builds a document collection for each social position. A social position's collection should contain documents with various topics that are relevant to the social position. This component is approached as a standard retrieval task. The social position is posed as a query to a web collection and top k documents are considered as a document collection for the social position. However, the use of the social position's terms as the only query terms would result in many irrelevant documents and such irrelevant documents would lead to a noisy representation. Therefore, I instead extracted seed terms, which act collectively towards the goal of guiding the representation process towards the relevant semantic space. The relevant semantic space here refers to the person sense of the social position. In the second component, I developed an LDA-based topic model to infer a topic as a representation for each social position using the social

[2] It is common for the hyperparameters α and β to be symmetric as in equation 4.9.

position's document collection. Section 4.2.1 details the process of building the document collection for each social position. In section 4.2.2, I present a differential LDA model (DiffLDA) as an extension to LDA to learn a social position's representation.

4.2.1 Building a document collection

4.2.1.1 Seeding

I hypothesised that different instantiations of the social position could act as disambiguation signals for the person sense of the social position. Each instantiation is constructed by prefixing an adjective to a social position. For example, *a leisure tourist*, *corporate tourist*, *British tourist,* and *rich tourist* are all *tourists.* This hypothesis might be supported by the results from my previous experiment in social position identification. In chapter 3, I found adjectives to be the single best performing feature set in identifying social positions. If adjectives are useful in identifying social positions, then they might also help in disambiguating, even partially, the *person* sense of social positions. The novelty in my approach to this subtask lies in the assumption that most instantiations of the social position would share a set of representative context terms with the original social position. In other words, we would expect terms like *baby*, *pregnancy*, *nutrition* and *kids* to be relevant to the social position *mother* as well as instantiations such as *single mother*, *divorced mother,* and *working mother.* Such terms would form the seed words to be used in building a richer representation for each social position.

Formally, I assumed a set of instantiations S for each social position and a set of adjectives J. I derive J from WordNet (Miller, 1995) and constructed $18,156$ instantiations for each social position. I further assumed a set of topical terms T for each social position. This set contained terms relating to the person sense of the social position and other terms that might represent different interpretations. For example, *tourist* may refer to a person or a movie, among other meanings. I expected terms such as *imdb*, *cast* and *trailer* that are likely to be related to the *movie* interpretation to be mixed with terms relevant to the person sense of the position such as *hotel*, *flight*, *booking,* and *attraction.* The task was then to identify a subset of T relevant to the social position.

There are several options to obtaining the set of topical terms T. I applied LDA (Blei et al., 2003) to the top 100 documents that were returned in response to submitting the social position as a query to a local search engine which I built during the course of this thesis. This search engine indexes category B of ClueWeb09 and uses a Dirichlet smoothed language model as its retrieval model. I also assumed a set of $R_{i,p}$ relevant documents for each instantiation i of each social position p. The set R is retrieved by posing the instantiation as a phrase query to the local search engine.

I used a simple voting mechanism to produce the final set of seed terms. This voting

Algorithm 3: Seeding algorithm

```
Input: A social position $p$ and a set of adjectives $J$
Output: A set of seed terms $E_p$
begin
    Preparation steps:
    foreach $j = 1...J$ do
        $S \longleftarrow$ Prefix $j$ to $p$ ;
    end
    $D_p \longleftarrow$ Retreive top documents for the query $p$ ;
    $T \longleftarrow$ Run LDA on the document collection $D_p$ ;
    foreach $i = 1...S$ do
        $D_i \longleftarrow$ Retreive top documents for instantiation $i$ ;
    end
    Voting steps:
    foreach $i = 1...S$ do
        foreach $d = 1...D_{i,n}$ do
            $V_{i,d} \longleftarrow$ Vote for $x$ terms $\in T$ ;
        end
        $V_i \longleftarrow$ Vote for $k$ terms by applying equation 4.10 on $V_{i,*}$ ;
    end
    $E_p \longleftarrow$ Vote for $n$ terms by applying equation 4.10 on $V_i$ ;
end
```

procedure operated in three steps: a document vote, an instantiation vote, and an aggregate vote. A relevant document $d \in R_{i,p}$ votes for at most x terms from the set of topical terms T. Relative frequency is used to rank topical terms in each document d. The top x terms are then considered as the terms voted for by document d. The second step is a vote at the instantiation level. Each instantiation i votes for at most k terms by using equation 4.10. This equation considers the relative frequency of documents $d \in R_{i,p}$ voting for a particular term, normalised by the total number of documents in the set $R_{i,p}$. The final vote aggregates the results of all instantiations voting, using equation 4.10, to produce n seed terms. Algorithm 3 provides a summary of this approach. As shown in the algorithm, the topics that are produced by LDA are pooled to form the set of topical terms T. I explored the option of choosing one of the topics that LDA produces as the seed representation of the social role using similar voting approaches. However, I found that most topics produced by LDA tended to contain extraneous terms and that pooling the terms as in algorithm 3 produces coherent seeds[3].

$$P(t) = \frac{\text{number of candidates voting in favor of term } t}{\text{total number of voters}} \tag{4.10}$$

[3] I set $x = n = k = 10$ empirically.

4.2.1.2 Retrieval

In general, topic models are applied to a document collection in order to represent the various topics that are covered in such a collection. In the previous section, I described a method to extract seed terms which are used to build a document collection for each social position from which the social position's representation is derived. The rationale behind this is that several semantic representations may exist, in the web, for each social position and it would be beneficial to ensure that the *person* sense of the social position is well represented in the document collection before applying any complex modelling technique. The seed terms provide such insurance and are practically considered as expansion terms in a document retrieval setting. I firstly constructed a weighted query that consisted of the social position's terms and the seed terms. Then, this query was submitted to a search engine and the top 100 documents are considered as the document collection[4]. The assumption is that such a query would return documents covering different topics and one of these topics is suitable to represent the social position. I further assumed that the relevant topic would be more likely to contribute in generating the top 10 documents. These top documents are considered as pseudo-relevant documents.

4.2.2 Differential Latent Dirichlet Allocation (DiffLDA)

I developed an extension to LDA (Blei et al., 2003), called DiffLDA, to probabilistically model a social position from its document collection. LDA assumes that a sparse mixture of topics generates each document and that each topic is a distribution over words. DiffLDA follows a similar assumption but accounts for three additional constraints: the social position topic is highly likely to contribute to generating pseudo-relevant documents; topics distinct from the social position's topic may also be found in the pseudo-relevant set; and the social position's topic is not restricted to the set of pseudo-relevant documents but can contribute to generating the other documents (i.e. documents at rank 11 to 100). These are soft constraints and may be overridden during estimation if enough evidence is found in the data. I further assumed the existence of a background topic that generates common terms in the document collection. Figure 4.4 presents the generative story of DiffLDA and Figure 4.3 shows the model as a plate diagram.

This model adds two components to the standard LDA model: the first is an observed regularisation distribution Ψ, and the second is a switching distribution Π in order to capture terms that are commonly found in the collection. The first component relates to my constraints regarding the social position's topic model and documents in the pseudo-relevance set. It is often the case that symmetric and low value priors are used when applying LDA to model documents, particularly to the α hyperparameter. The low value

[4] I use a local search engine over ClueWeb09 category B.

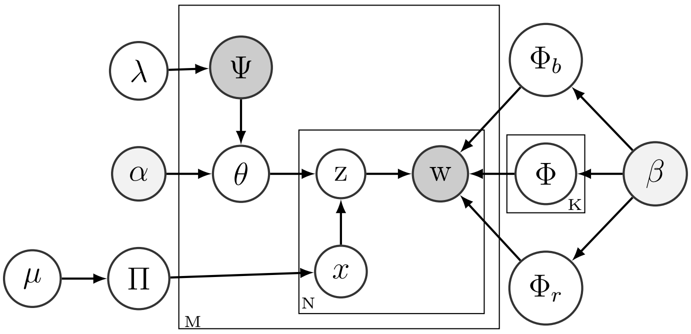


Figure 4.3: A graphical representation of DiffLDA.

```
Draw a switching prior distribution $\Pi \sim Dir(\mu)$
For each topic $k \in \{1, ..., k\}$ :
   Draw a topic distribution $\Phi_k \sim Dir(\beta)$
Draw a background topic distribution $\Phi_b \sim Dir(\beta)$
Draw a position topic distribution $\Phi_r \sim Dir(\beta)$
For each document $m$:
  Draw $\Psi^t_m \sim Bernoulli(\lambda)$
  Generate $\boldsymbol{\alpha_m} = \boldsymbol{L_m} + \boldsymbol{\alpha}$
  Draw document topic distribution $\Theta_m \sim Dir(\alpha_m)$
  For each word $n$ in document $m$:
     Draw a switching variable $x_{m,n} \sim Multi(\Pi, 1)$:
     If $x_{m,n} = background$ :
     Draw a word $w_{m,n} \sim Multi(\Phi_b, 1)$
     If $x_{m,n} \neq background$ :
     Draw a topic $\Phi_t \sim Dir(\beta)$
     Draw a word $w_{m,n} \sim Multi(\Phi_t, 1)$
```

Figure 4.4: The generative process of DiffLDA

of such a parameter encourages the model to assign few topics for each document. When modelling social positions representations, it would be preferable to bias the document topic distribution θ to include the social position's topic as one of the topics for each document in the pseudo-relevance set. Note that I prefer not to force the model to include the position's topic but more to encourage the inclusion of this topic unless empirical evidence suggests otherwise. More formally, let R be the set of pseudo-relevant documents and let $\boldsymbol{L}$ be a vector of length K where $\boldsymbol{L}^k_m$ refers to the topic k entry in vector $\boldsymbol{L}$ of document m. Similarly, $\boldsymbol{L}^r_m$ refers to the social position's topic, r, entry in vector $\boldsymbol{L}$ of document m. I set this vector as follows:

- $\boldsymbol{L} = \mathbf{0}$ if document $m \notin R$. $\mathbf{0}$ is a vector of all zeros of size K, effectively keeping the hyperparameter α unchanged.

- $\boldsymbol{L} = \boldsymbol{u}$ if document $m \in R$. $\boldsymbol{u}$ is a vector of all zeros of size K except at $\boldsymbol{u}_r = \tau$, where r is the index of the social position's topic.

The parameter τ acts as a bias parameter that increases the probability of selecting the social position's topic for documents in the pseudo-relevant set. I set $\tau = 2\alpha$. Both hyper-parameters α and β are optimised using a maximum likelihood estimation provided by the Mallet library (McCallum, 2002). The document topic hyperparameter α is then set according to step 8 in the generative process for each document m (Figure 4.4). A similar transformation is used in the laballed LDA topic model (Ramage et al., 2009) where α is projected into a lower dimensional vector to restrict LDA to predefined topics obtained from manual annotation.

4.2.2.1 Inference

I use a collapsed Gibbs sampling (Griffiths and Steyvers, 2004) technique to infer the latent variables x and z. The full joint probability for the model in figure 4.3 is as follows:

$$\begin{aligned} P(\boldsymbol{\Pi}, \boldsymbol{\Psi}, \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\Phi} | \mu, \lambda, \alpha, \beta) = {} & \underbrace{P(\boldsymbol{\Pi}|\mu)}_{\text{Dirichlet}} \times \underbrace{P(\boldsymbol{\Psi}|\boldsymbol{\lambda})}_{\text{Bernoulli}} \times \underbrace{P(\boldsymbol{\theta}|\alpha, \boldsymbol{\Psi})}_{\text{Dirichlet}} \times \underbrace{P(\boldsymbol{x}|\Pi)}_{\text{Categorical}} \\ & \times \underbrace{P(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})}_{\text{Categorical}} \times \underbrace{P(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\Phi})}_{\text{Categorical}} \times \underbrace{P(\boldsymbol{\Phi}|\beta)}_{\text{Dirichlet}} \end{aligned} \tag{4.11}$$

Equation 4.11 can be re-written as follows:

$$\begin{aligned} P(\boldsymbol{\Pi}, \boldsymbol{\Psi}, \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\Phi} | \mu, \lambda, \alpha, \beta) = {} & P(\boldsymbol{\Pi}|\mu) \times \prod_{m=1}^{M} P(\boldsymbol{\Psi_m}|\lambda) \times \prod_{m=1}^{M} P(\boldsymbol{\theta_m}|\alpha, \boldsymbol{\Psi_m}) \\ & \times \prod_{m=1}^{M} \prod_{n=1}^{N} P(\boldsymbol{x_{m,n}}|\Pi) \times \prod_{m=1}^{M} \prod_{n=1}^{N} P(\boldsymbol{z_{m,n}}|\boldsymbol{x_{m,n}}, \boldsymbol{\theta_{m,n}}) \\ & \times \prod_{m=1}^{M} \prod_{n=1}^{N} P(\boldsymbol{w_{m,n}}|\boldsymbol{z_{m,n}}, \boldsymbol{\Phi_{z_{m,n}}}) \times \prod_{k=1}^{K} P(\boldsymbol{\Phi_k}|\beta) \end{aligned} \tag{4.12}$$

The latent variables in the model are: θ, Π, x, z and Φ. Collapsed Gibbs sampling (Griffiths and Steyvers, 2004) works by integrating out the variables θ, Π and Φ. The joint probability would then be simplified as in equation 4.13 from which samples would be drawn.

$$P(x, z, w) = P(x) \times P(z|x) \times P(w|z) \tag{4.13}$$

I sample x and z jointly by alternating between the two different cases in step 12 and 14 in the generative process (Chemudugunta et al., 2006). If the switching variable $x_{m,n}$ turns out to be background, then the word is drawn from the background topic as in equation 4.14. The second case refers to the situation where $x_{m,n}$ is not a background term in which case probability estimation is based on equation 4.15.

$$P(x_i = b, z_i = b, w_i = w_i) \propto \frac{c(d_i, b) + \mu_b}{c(d_i, *) + \sum_x \mu_x} \times \frac{\Delta^{\neg i}_{w_i,b} + \beta}{\Delta^{\neg i}_{*,b} + V\beta} \tag{4.14}$$

$$P(x_i = \neg b, z_i = t, w_i = w_i) \propto \frac{c(d_i, \neg b) + \mu_{\neg b}}{c(d_i, *) + \sum_x \mu_x} \times \frac{\Omega^{\neg i}_{d_i,t} + \alpha_t}{\Omega^{\neg i}_{d_i,*} + \sum_t \alpha_t} \times \frac{\Delta^{\neg i}_{w_i,t} + \beta}{\Delta^{\neg i}_{*,t} + V\beta} \tag{4.15}$$

where $c(d_i, b)$ and $c(d_i, \neg b)$ are simply the counts of tokens in the background topic and tokens not in the background topic in document d_i, respectively. $c(d_i, *)$ is the number of tokens in document d_i and V is the number of words in the vocabulary. Ω and Δ are count matrices as the following:

1. Ω is a matrix of dimension $D \times T$, where D is the number of documents and $T = K + 1$ is the number of topics; including the social position topic. $\Omega^{\neg i}_{d_i,t}$ is the number of times topic t is assigned to document d_i with the assignment of the i^{th} word excluded.

2. Δ is a matrix of dimension $V \times N$, where $N = T + 1$ to account for the background topic. $\Delta^{\neg i}_{w_i,t}$ is the number of times term w_i is assigned to topic t with the assignment of the i^{th} word excluded.

4.3 Experimental setup

In this section, I describe the settings and evaluation metrics that were used to provide an intrinsic evaluation of the user models produced by the DiffLDA model. A social position's model is represented as a topic, which is composed of a group of words that are related to each other (Boyd-Graber et al., 2017). One way of evaluating such models is to measure their perplexity on a held-out document collection (Wallach et al., 2009). In the context of this thesis, such models are only the starting point of representing a social position. Each social position might reasonably have a wide range of interests and, therefore, such models would be configured and extended accordingly depending on the task. The role of such a user model is to identify words that are relevant to a particular social position and make sense as a group. Chang et al. (2009) asked human annotators to evaluate the coherence of topics and concluded that human judgments of coherence correlate negatively with traditional metrics such as log odds and predictive likelihood. A more appropriate evaluation approach, in the context of this thesis, would be to quantify the interpretability or coherence of such models.

Newman et al. (2010) explored various automatic measures of coherence utilising structural knowledge of resources such as WordNet (Miller, 1995) or Wikipedia. They have also investigated term co-occurrence using the pointwise mutual information (PMI) measure between each pair of topic words as in the following:

$$PMI(w_i, w_j) = log\frac{P(w_i, w_j)}{P(w_i)P(w_j)} \tag{4.16}$$

The coherence score for a topic is the average PMI of all possible pairs of the top n words for that topic. Newman et al. found that such a PMI-based measure correlated well with human annotators. Follow-up studies (Aletras and Stevenson, 2013; Lau et al., 2014) found that using the normalised PMI (NPMI) (Bouma, 2009) improved correlation with human annotators further. NPMI is calculated as follows:

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-log(P(w_i, w_j))} \tag{4.17}$$

Mimno et al. (2011) also used co-occurrence statistics at the document level to compute topic coherence. They used log conditional probability instead of PMI and also found correlation with human judgments. Aletras and Stevenson (2013) suggest building distributional vector representations for each word and then applying various metrics to measure topic coherence. Chang et al. (2009) introduced another method of evaluating coherence called *word intrusion* where an annotator would be presented with the top 9 words of a topic with an additional randomly sampled word in random order. The task is to identify the word that does not belong to the topic. Identification of the intruder word would be easy if the topic is coherent. Lau et al. (2014) introduced an approach to automate word intrusion evaluation.

In this chapter, I use the NPMI and PMI metrics to evaluate the topic coherence of social positions' models. Following Lau and Baldwin (2016), I also report the average NPMI and PMI over four topic cardinalities: 5, 10, 15, and 20. The main approach of this chapter is the DiffLDA model. I compare its output to a baseline method that is based on relevance modelling (Lavrenko and Croft, 2001). Each social position is submitted to a local search engine over ClueWeb09 category B. A relevance model is then estimated over the top 10 documents. The top n words of the relevance model are considered as a topic. Similar to Newman et al. (2010), I use a Wikipedia dump to calculate co-occurrence scores. Statistical significance is measured using paired t-test ($p < 0.05$).

4.4 Results

Table 4.1 presents the coherence scores for the DiffLDA and relevance modelling approaches across four topic cardinalities using the NPMI and PMI metrics. DiffLDA provided a statistically significant improvement over the relevance modelling approach on both metrics and for all considered sizes. This result suggests that the DiffLDA model, in general,

Topic cardinality	5	10	15	20	Avg
	NPMI				
Relevance modelling	0.1048	0.0638	0.0480	0.0387	0.0640
DiffLDA	**0.1776***	**0.1340***	**0.1102***	**0.0956***	**0.1294***
	PMI				
Relevance modelling	0.5057	0.2984	0.2213	0.1741	0.3000
DiffLDA	**0.8288***	**0.6362***	**0.5288***	**0.4621***	**0.6140***

Table 4.1: Topical coherence results. (*) indicates statistical significance ($p < 0.05$).

Social position	**Top words**	**NPMI**
Wine lover	wine sauvignon grape pinot cabernet noir blanc taste bordeaux chardonnay	0.36
Organ donor	transplant marrow organ bone donor cell blood patient disease stem	0.28
Internal medicine physician	medical medicine physician health nursing surgery care pediatric nurse clinical	0.28
Pathologist	pathology medicine pathologist medical surgery disease clinical patient cancer care	0.27
Astronaut	space nasa mission astronaut mars moon shuttle orbit launch apollo	0.26

Table 4.2: DiffLDA example topics.

successfully learns coherent sets of terms to represent social positions. The reduced performance of the relevance modelling approach could be attributed to the fact that such a model assumes that the top k documents are relevant to the social position which may not hold in practice. Irrelevant documents might be present among the top k and contribute noisy terms to the relevance model. In contrast, DiffLDA uses a seeding approach to generate small and potentially representative terms for each social position. Seed terms are used to generate a document collection for each social position from which DiffLDA learns a model for each social position. Tables 4.2 and 4.3 present examples of the most coherent topics that are produced using the DiffLDA and relevance modelling, respectively.

To examine the effect of the seed terms on the performance of DiffLDA, I calculated the NPMI coherence score for the seed terms and measured the Pearson's correlation coefficient between the seed and the DiffLDA scores. Since the number of seed terms for each social position is 10, NPMI scores were calculated based on this topic size. A strong correlation ($r = 0.554$) was found, which suggests that DiffLDA performance depends on the seed terms. Interestingly, a correlation ($r = 0.3620$) was also found between DiffLDA and the relevance modelling approach. This correlation, although moderate, might indicate that a coherent set of terms is perhaps possible to obtain for a particular set of social positions. For example, social positions like *Clinton supporter, Dietitian, Diabetic, bicyclist, iPhone user,* and *Peace activist* produce coherent topics using both approaches. In contrast, social

Social position	Top words	NPMI
Science graduate	graduate science school program arts politics degree study university college	0.22
Roman catholic	roman catholic diocese cathedral church saint roma st archdiocese catholicism	0.22
Marxist	party communist marxist leninist india committee worker nepal revolutionary central	0.22
Mental patient	patient mental health care treatment disorder service hospital medical illness	0.22
Evolutionary biologist	evolutionary biology biologist evolution geneticist molecular dead genetics human study	0.21

Table 4.3: relevance modelling example topics.

positions such as *web developer, movie fan, travel agent, art lover,* and *bird watcher* were among the lower quarter for both approaches based on NPMI scores. This is perhaps a limitation for the coherence evaluation metrics. By definition, coherence assumes that the topic terms would often co-occur with each other. This might be the case for narrow social positions that are based on a single concept, person, or product but not for broad positions.

4.5 Summary

The previous chapter introduced a framework to model search engine users based on the concept of social positions. The focus was on extracting and identifying social positions from web documents. In this chapter, I built a representation based on a topic modelling approach for each identified social position using web documents. Each social position was modelled as a multinomial distribution over words, i.e. a topic. In section 4.1, I presented two prominent topic modelling approaches: LSA and LDA. Section 4.2 detailed the formulation of this representation task. There were two main phases of this chapter's approach: building a document collection and estimating a social position's topic using a differential LDA model. These two components were described in sections 4.2.1 and 4.2.2, respectively. Sections 4.3 and 4.4 presented the experimental settings and results. I demonstrated that semantically coherent representations of social positions could be learnt in unsupervised settings and using publicly available web documents. In chapters 5 and 6, I use the top 2000 social positions and their learned models.

CHAPTER 5

Framework validation I: Search results diversification

Throughout chapters 3 and 4, I presented a framework for building and maintaining user models from web documents. In this chapter, I evaluate the utility of these user models using the task of search results diversification. Diversification algorithms aim to present search results to the satisfaction of users with diverse interests. It is often suggested as one of the preferred techniques for dealing with search query ambiguity. By presenting results that are relevant to more than one interpretation, the search engine will have a better chance of satisfying users with varying interests. The diversification task constitutes a suitable extrinsic test collection for the user models I built in the previous chapter. To be able to diversify search results, the different interpretations behind the search query must be modelled. That is, there is a pre-requirement for diversification algorithms to have an understanding of the multiple possible intents behind the search query. The developed user models aim to provide such an understanding. By identifying social positions for each web document returned as a result for a search query, a generalised understanding of the different audiences who might have an interest in the submitted search query can be achieved. Thus, the role of such user models is to provide the diversification algorithm with representations of the different users who might have submitted the query.

This chapter is structured as follows. Section 5.1 discusses the issue of query ambiguity as a source of empirical motivation for diversification. In section 5.2, previous diversification approaches are reviewed. Test collections and metrics that are developed to evaluate the diversity and novelty of search results are discussed in section 5.3. The main contribution of this chapter is a diversification method based on social positions which is described in section 5.4.

5.1 Motivation

Users' information needs are conveyed through search queries. Traditionally, a search query is considered as a representation for a single information need (Spärck-Jones et al., 2007). In practice, numerous empirical findings suggest that search queries may not fully and unambiguously express information needs. One factor that possibly leads to ambiguity in a search query is its length. Analysis of query logs from multiple search engines such as AOL (Pass et al., 2006), AltaVista (Silverstein et al., 1999), Excite (Jansen et al., 2000; Lau and Horvitz, 1999; Spink et al., 2001), and Microsoft Live Search (Zhang and Moffat, 2006) have all shown that search queries are typically short, ranging between two to three terms on average. The fewer terms the user provides to communicate his/her need then the more likely that the search query is going to be ambiguous.

There are different types of ambiguities related to search queries with the first being word sense (Sanderson, 1996). In daily life, it is quite easy for people to determine the correct sense of an ambiguous term based on its first few surrounding words (Choueka and Lusignan, 1985; Miller, 1951). Nearby terms, or contextual signals, are also influential in automatic disambiguation algorithms (Navigli, 2009). However, it is challenging to decide which sense the user has in mind when submitting a short query as the contextual signals are sparse or do not exist. Thus, as can be expected, shorter queries are those that are most affected by sense ambiguity (Sanderson, 1996). Query aspect represents the second type of query ambiguity. A query that has a clear interpretation might still be ambiguous at the aspect level. For example, the query *Harry Potter* might mainly refer to the fictional character but remains ambiguous because it is not clear whether the user is interested in *Harry Potter books*, *movies*, *costume,* or *songs.* This type of ambiguity is referred to as *underspecified queries.*

Thirdly, a search query can be classified as ambiguous at the request type level (Spärck-Jones et al., 2007). According to Broder (2002), there are three main types of search queries: informational, navigational, and transactional. An informational query refers to a user's quest to acquire information about a particular topic from one or multiple resources. If the user is interested in reaching a specific website, then the query is considered to be navigational. Transactional queries, or resource seeking queries as described by Rose and Levinson (2004), are those requests that indicate the user's interest in acquiring a resource for further interaction. Examples of transactional queries include *downloading a song* or *watching a movie online.*

There are several proxy approaches to quantifying the extent of query ambiguity in web search. Dou et al. (2007) used clickthrough data to define a measure of variability among users when issuing the same query based on information entropy as in equation 5.1:

$$clickEntropy(q) = \sum_{d \in D} - P(d|q)\ log_2\ P(d|q) \tag{5.1}$$

D is the collection of web pages that have been clicked on for query q. A smaller click entropy indicates that users tend to agree on which documents to click on. This might signal that the query is less ambiguous. Dou et al. found that the majority of popular queries have low click entropies. Their results also showed that personalisation algorithms improve search results significantly for queries that have large click entropy. One issue with the click entropy measure is that it fails to distinguish between ambiguous and informational queries. An informational query might be associated with many documents in the clickthrough logs and thus will have high entropy regardless of being ambiguous or not.

Wang and Agichtein (2010) sampled queries from the Microsoft Live Search logs and manually labelled them into three categories: clear, informational, and ambiguous. The clear category refers to unambiguous navigational queries. Queries were divided into three groups: low ($10 - 100$ clicks), medium ($100 - 1000$ clicks), and high (over 1000 clicks). They then sampled 50 queries per group for manual labelling. Queries in the high group were mostly navigational (76%) and less ambiguous (16%) compared with those in the medium (28% ambiguous) and low frequency groups (22% ambiguous). Song et al. (2009) presume that analysing the query's top returned documents can identify ambiguity. Their assumption is that an ambiguous query will return documents that belong to different topic categories in a pre-defined topic taxonomy. In their study, an ambiguous query is one that has multiple topical interpretations. They estimated that 16% of queries are ambiguous using a topic classifier. It is worth noting that underspecified queries were considered as *broad queries* rather than being ambiguous so such a figure might be a conservative estimate of the true proportion of ambiguous queries in query logs.

Proper nouns such as people's names, cities, products, or acronyms are another source of ambiguity in web search. Sanderson (2008) studied the prevalence of this type of ambiguity using query logs from a web search engine. In the study, a query was judged as being ambiguous if it has a Wikipedia disambiguation section or has multiple entries in WordNet. About 16% of the most frequent queries were found to be ambiguous[1]. This study further concluded, via a simulated retrieval task, that the retrieval performance of an IR system was negatively affected in the presence of ambiguous queries. Clough et al. (2009) conducted a query log analysis study following a similar methodology of identifying ambiguous queries based on WordNet and Wikipedia disambiguation pages. They found that queries with multiple interpretations did not correlate with a higher click entropy

[1] Frequent queries are those that appeared 87 times or more in the query logs. If all queries are considered, the percentage of ambiguous queries would be 3.9%.

value as would have been expected. In fact, widely used queries illustrating the issue of ambiguity in query logs such as *Jaguar* or *Java* were found to have relatively low entropies (2.73 and 3.73 respectively). Instead, the click entropy for a query correlated positively with the length of the query's article on Wikipedia[2]. The length of the article was taken as a signal for the breadth of the query topic and that such queries might have possibly comprised several subtopics. They also estimated that 16.20% of queries in Microsoft Live Search logs had either high entropy value or had been reformulated by the user at least once. Such queries would potentially benefit from an approach to deal with ambiguity.

Teevan et al. (2007b) examined the variability in explicit relevance judgments provided by different users for identical queries. Users were asked to select queries from a set provided by the researchers. Participants would then write a detailed description of the information need that they think each query would express and use their own description to judge the relevance of results. They observed a low inter-rater agreement (56%) between users evaluating the same queries. Hafernik and Jansen (2013) manually classified 5,115 queries into two categories: specific and general. A query is considered specific if it has one of nine attributes such as: contains a URL[3], contains a place name or location with additional terms, contains a question with a clear answer, or contains a name with additional terms. They found that 38% of queries were general. In addition, specific queries were twice as long as general queries (4.5 vs 2.1 terms). Phan et al. (2007) conducted user studies to investigate the correlation between query length and query specificity or broadness. They found that broad queries tended to be, on average, less than 3 terms.

5.2 Background

Web search is an interactive process between a user and a search engine. The search engine facilitates such an interaction through a user interface, which provides a possible venue to address challenges such as ambiguity. However, the user interface to web search engines has traditionally been simple; a user submits a query and the search engine responds with a ranked list of results. Several other search interfaces have been suggested in the literature that are based on techniques such as clustering (Zamir and Etzioni, 1999), categorisation (Chen and Dumais, 2000), or visualisation (Hoeber and Yang, 2006). These advanced interfaces seek to help the user navigate the information space but require additional effort from the user to make his/her search intent more salient. Despite the potential improvements that such interfaces could add to the search process, the simple ranked list remains dominant. One reason for the popularity of such a simple interface is that web

[2] This was calculated for ambiguous queries that have a dominant interpretation. The dominant interpretation is provided by the Wikipedia community.

[3] In fact, not all URL queries should be considered navigational or unambiguous. Lee and Sanderson (2010) show that about 14% of URL queries were not navigational.

search is a supporting task rather than a goal in itself (Hearst, 2009). Users might not want to exert the additional cognitive effort that complex search interfaces might require compared with the easy and well-understood ranked list interface (Hearst, 2009). Thus, different approaches to dealing with web search challenges have been introduced within the traditional ranked-list interface. An example of which is search results diversification.

Before presenting the different technical views of diversification, it is worth re-stating one influential concept based on which most ranked-lists of documents are generated. This concept underlines probabilistic IR and is known as the Probability Ranking Principle (PRP)[4] (Robertson, 1977). It states that the *ideal* response of an IR system to a user's request would be a ranked-list of documents in the order of their probability of relevance to the user's request given all the evidence available to the IR system. Retrieval models that are probabilistic such as BM25 (Robertson and Walker, 1994) or those based on language modelling such as QL (Ponte and Croft, 1998) follow this ranking policy. This principle, however, implies some simplifications that may not make the system's response effective from the user's point of view.

Firstly, the PRP assumes that each request represents a single information need and that documents should be ranked based on their relevance to that need (Spärck-Jones et al., 2007). Empirically, however, queries do not always express a single information need and ambiguity in search queries is common as discussed in the previous section. If a query has two interpretations, a result list that ranks optimally for one of the interpretations is of little value to the user interested in the other interpretation. Secondly, the PRP presumes that the relevance of a document to a query is independent of other documents in the retrieved list (Croft et al., 2010). This document independence assumption does not hold in practice. For instance, Craswell et al. (2008) presented a model called *the cascade model* that attempts to explain how users behave with respect to examining the result list. The model suggests that users view documents from top to bottom and leave when a helpful document is found. In this model, the probability of clicking on results at lower ranks decreases when a document at the top has been clicked on. Once the user finds a document with the sought after information need; the relevance of any other document at a lower rank with duplicate information becomes questionable. Also, a document relevance to a query for a particular user is affected by many external variables that are not likely to be known to the IR system and therefore the system's result list is unlikely to be the ideal response (Baeza-Yates and Ribeiro-Neto, 2011).

Diversifying search results provides possible solutions for the above shortcomings of the PRP. Firstly, the assumption that each query represents *a single* information need can be replaced by one that considers *multiple* needs behind each query. Diversification in this context would provide a result list with *maximum coverage* of the multiple possible needs.

[4] The original PRP statement is quoted in chapter 2.

Secondly, the document independence assumption which does not penalise *redundancy* in the results list would be overridden by one that does. In such a case, diversification might be achieved by promoting *novelty*, or minimising redundancy, in the results list. Novelty-based diversification is tightly linked to the cascade user-browsing model (Craswell et al., 2008). If the user examines documents sequentially from top to bottom, then the average rank at which users with diverse interests would find a relevant document should be minimised. Coverage-based methods do not seek to optimise such an objective. Instead, the objective is to cover the multiple possible needs not necessarily to minimise the average rank at which a relevant document is found. Following the TREC diversification track, I use the term subtopic, denoted with the letter t, to refer to a specified version of an ambiguous query. A subtopic may indicate the intended interpretation of an ambiguous query or an aspect of a multi-facet query. For example, *java programming* is a subtopic that communicates the intended sense for the more general query *java*. Similarly, *Harry Potter books* is a subtopic for *Harry Potter* focusing on *books* related information. Throughout this chapter, I use the symbol $T(q)$ to denote the set of subtopics for query q.

In the following, I provide a review of coverage-based approaches in section 5.2.1 followed by novelty-based diversification in section 5.2.2. Section 5.2.3 highlights other approaches to diversification.

5.2.1 Coverage-based approaches

Agrawal et al. (2009) formulated an objective function aimed to maximise the probability that the average user will find a relevant document in a result list of size k. This work is based on the assumption that users will examine the top k documents. Note that this objective does not directly optimise the rank at which the average user will find a relevant document but merely that the result list includes a relevant document. Their Intent Aware Selection (IA-Select) algorithm relies on topic categories to represent the different possible intents behind a search query. Each document might belong to one or several topic categories and as do search queries. The objective function is as follows:

$$P(S|q) = \sum_{c} P(c|q)(1 - \prod_{d \in S}(1 - V(d|q,c))) \tag{5.2}$$

where S is the set of selected documents of size k. $P(c|q)$ is the probability that query q belongs to the topic category c and is estimated based on query logs of a commercial search engine (Fuxman et al., 2008). $V(d|q,c)$ quantifies the relevance of document d to query q with respect to topic category c. The above objective function does not aim to cover the different topic categories of a search query proportionally to their estimated relatedness $P(c|q)$. Instead, it places more weight on finding highly relevant documents within each category. Also, redundancy is not directly accounted for since the relevance of a document to a specific query is independent from any other documents within the same topical category.

Carterette and Chandar (2009) proposed a probabilistic set-based approach for the special case of diversification where the *correct* interpretation of the query can be assumed. Their model attempts to generates a result list that covers as many subtopics of the query as possible according to the following formula:

$$P(T \in D) = \prod_{m=1}^{M} P(t_m \in D) = \prod_{m=1}^{M} 1 - \prod_{i}^{n}(1 - P(t_m|d_i)) \tag{5.3}$$

Their goal is to maximise the number of subtopics T that the result list D covers. Three optimisation strategies were used to achieve this goal. The best performing strategy operates by selecting the best document d_i for each subtopic t_m based on $P(t_m|d_i)$. The selected documents are then re-ranked based on their topical relevance to the query. Subtopics were generated using LDA or a relevance modelling method. Both methods would naturally provide estimates for $P(t_m|d_i)$, which is needed for the selection step. Although this approach does not penalise redundant documents, it achieves novelty by selecting documents that contain *new* subtopics that have not yet been covered in the selected set of documents. The IA-Select algorithm, on the other hand, does not model the information contained within a candidate document with respect to the documents already selected. If the algorithm decides to include a document from a previously visited topic category, the new document might not necessarily provide a *new* piece of information.

He et al. (2011) also applied LDA to obtain query-specific clusters. These clusters were then ranked based on the probability of each cluster generating the search query $P(t|q)$, which is derived using the topic model for each cluster. The resulting clusters, and their $P(t|q)$, formed topic representations of the search query and were provided as inputs to diversification algorithms such as IA-Select or the set-based probabilistic approach of Carterette and Chandar (2009). The best performing diversification approach was based on a round robin selection strategy. In each round, a document is selected from the top J clusters based on its relevance to the query and added to the rank list.

Radlinski and Dumais (2006) explored search results diversification in the context of client-side search results personalisation. In client-side personalisation, the top k search results are provided by the search engine and then re-ranked according to the user's profile. The re-ranking process is performed at the user device. Their goal was to ensure that the top k documents are diverse so that different users would find relevant documents at the top when re-ranking is performed. They obtained the different possible subtopics of each query utilising query reformulations from a commercial search engine. Their diversity function forms the top k results by selecting an equal number of documents from each subtopic as well as the original query and appends them to the list.

Rather than proportionally allocating documents based on the number of subtopics as in the Radlinski and Dumais (2006) study, Dang and Croft (2012) diversified search results proportionally based on subtopics popularity. Their work is analogous to the problem of seat allocation in elected parliaments. Each position in a search result list is viewed as a seat in parliament and each subtopic is a party. Their greedy algorithm, called PM-2, implements diversification by ensuring that the diversified search results list proportionally represents the popularity of the query's subtopics. PM-2 makes three decisions at each position in the ranked-list. First, it calculates the *quotient*[5], a score used to rank subtopics based on their popularity and how well they are already represented in the current results list. The formula to calculate the quotient is as follows:

$$quotient[t] = \frac{v_t}{2s_t + 1} \tag{5.4}$$

s_t is the number of positions occupied by documents relevant to subtopic t while v_t is the popularity of subtopic t. Secondly, PM-2 selects the best document to fill the current search position as follows:

$$d^* \leftarrow \arg\max_{d \in \mathcal{D}} \left(\lambda \times quotient[t^*] \times P(d|t^*) \right) + \left((1-\lambda) \times \sum_{t \neq t^*} quotient[t] \times P(d|t) \right) \tag{5.5}$$

It is often the case that search results diversification is formulated as a linear interpolation between a relevance function and a novelty function. In other words, there is a trade-off between promoting a highly relevant document or a *novel* document. Retrieval models are well-studied in IR and functions such as the QL (Ponte and Croft, 1998) or the BM25 (Robertson and Walker, 1994) can be used to estimate the document's relevance to a query, as reviewed in chapter 2. The novelty function, however, depends on how the diversification function is defined and this is where most of the previous studies on diversification algorithms differ. This trade-off applies to both novelty-based and coverage-based methods. PM-2 trades using λ between the relevance of the document d to subtopic t^* which is the highest ranked subtopic according to equation 5.4 and the relevance of the document to the other subtopics. In the final step, the variable s_t is updated to account for how much the selected document covers the different subtopics as follows:

$$s_t \leftarrow s_t + \frac{P(d^*|t)}{\sum_{t \in T(q)} P(d^*|t)} \tag{5.6}$$

Liang et al. (2014a) used a variant of the PM-2 approach to select a document from multiple lists of results generated for the same search query. Most previous studies approach

[5] Their algorithm is based on the Sainte-Laguë method which is a seat allocation formula used in several political institutions.

diversification as a re-ranking task. Liang et al., instead, studied diversification as a fusion task. Submitted runs to the TREC diversification task were considered as candidate lists for a fusion function to generate a diversified list. They used an LDA-based topic model to infer the latent subtopics of each query from the candidate ranked lists. The probability that a query is generated by a particular document is then calculated as follows:

$$P(d|q) \approx F_{fusion}(d|q) \tag{5.7}$$

where $F_{fusion}(d|q)$ is calculated using the *CombSUM* fusion formula due to Shaw and Fox (1994).

5.2.2 Novelty-based approaches

In this section, I review approaches that explicitly account for redundancy. This is usually achieved by comparing the current document with documents already selected using a novelty function. Carbonell and Goldstein (1998) presented Maximal Marginal Relevance (MMR) as one of the earliest novelty functions. MMR discounts a document if a similar document has already been added to the diversified list. Documents are selected according to the following equation:

$$d^* \leftarrow \arg\max_{d \in \mathcal{D} \setminus D} \Big(\lambda \times P(d|q) \Big) - \Big((1-\lambda) \arg\max_{\hat{d} \in D} F_{sim}(d, \hat{d}) \Big) \tag{5.8}$$

$\mathcal{D}$ is the set of documents returned as a result of the search query. D is the currently selected documents. The first document to be selected is the most relevant document based on $P(d|q)$. For the remaining positions in the list, MMR will trade between document relevance and its similarity to selected documents using λ. F_{sim} is a similarity function such as the cosine similarity. Zuccon and Azzopardi (2010) introduced a parameter-free subtopics retrieval model which is similar to MMR except that it uses quantum interference to quantify the novelty of a document. Zhai et al. (2003) presented a probabilistic extension of MMR in which documents are selected according to the following equation:

$$d^* \leftarrow \arg\max_{d \in \mathcal{D} \setminus D} P(d|q) \Big(1 - \lambda - F_{MIX}(d, \theta_D) \Big) \tag{5.9}$$

where $P(d|q)$ is the KL-divergence relevance score. θ_D is a language model of the documents that are already in the list D. $F_{MIX}(d, \theta_D)$ attempts to quantify how much of the current document is generated from the language model θ_D or a background English language model by estimating the co-efficient μ for the mixture model. It is calculated as follows:

$$F_{MIX}(d, \theta_D) = \arg\max_{\mu} \mathcal{L}(\mu | d, \theta_D) \tag{5.10}$$

Zhang et al. (2005) constructed an affinity graph between all documents in the retrieval collection. If the content similarity score between document d_i and d_j is bigger than a pre-defined threshold, a directional edge from d_i to d_j is created with the similarity score being the edge's weight. An information richness score is then calculated for every document using a link analysis algorithm. Finally, documents are ranked based on a linear combination of their relevance to the query and their information richness score. To promote diversity at each position in the search results list, the information richness scores of documents are discounted if a neighbouring document has already been added to the list. Gollapudi and Sharma (2009) explored two different methods of defining a similarity function between two documents. The first one calculates distance using Jaccard similarity of documents sketches built using a min-hashing scheme (Broder et al., 2000) while the second one uses a measure of topical category distance between documents. Chen and Karger (2006) developed a diversity model with the goal of including at least *one* relevant document in the results list. To achieve this goal, they developed a probabilistic model in which the relevance of a document was conditioned on the assumption that all previous documents were irrelevant to the user need.

Santos et al. (2010) suggested a probabilistic function, called xQuAD, that promotes diversity by decomposing a search query into a set of subtopics. Instead of comparing documents with each other to determine the novelty of a document, xQuAD uses subtopics to determine the novelty of a document. It considers a document novel if it is relevant to the query and covers subtopics that are not already covered in the results list. Documents are selected according to the following function:

$$d^* \leftarrow \arg\max_{d \in \mathcal{D}} \Big(\lambda \times P(d|q) \Big) + \Big(1 - \lambda \times \sum_{t \in T(q)} P(t|q) \times P(d|t) \times \prod_{\hat{d} \in D} 1 - P(\hat{d}|t) \Big) \tag{5.11}$$

λ is a parameter that controls the trade-off between relevance and diversity. $p(t|q)$ is the importance of subtopic t given the search query q. Zheng et al. (2012) explored different techniques to define an objective function that maximises the diversity of a result list. The best performing objective function is similar to the above xQuAD function except that it selects a document d based on a square-loss function as follows:

$$d^* \leftarrow \arg\max_{d \in \mathcal{D}} \Big(\lambda \times P(d|q)\Big) + \Big(1 - \lambda \times \sum_{t \in T(q)} P(t|q) \times P(d|t) \times (2 - 2 \sum_{\hat{d} \in D} P(\hat{d}|t) - P(d|t))\Big) \tag{5.12}$$

Intuitively, this function also discounts the novelty of documents covering subtopics that had already been covered in the rank list. They used a Dirichlet-smoothed language model (Zhai and Lafferty, 2004) to estimate $P(d|q)$ in the relevance function as well as $P(d|t)$ and $P(t|q)$ in the diversification function. One of the objective functions explored by Zheng et al. was a summation-based function introduced by Yin et al. (2009) as follows:

$$P(d) = \sum_{t \in T(q)} P(d|t)P(t|q) \tag{5.13}$$

They found that this function performed poorly compared with the square loss function discussed above. Its performance is possibly diminished for two reasons. Firstly, diversification is usually modelled as a trade-off between document relevance and coverage or novelty. This function ignores the document relevance to the query, which tends to empirically outweigh coverage or novelty in a trade-off function. Secondly, this summation function does not estimate how much of a particular subtopic t has already been covered in the result list, which essentially makes it a coverage-based function.

5.2.3 Other diversification approaches

More recently, machine learning approaches have been used to learn a ranking function that takes into account multiple relevance and diversity features rather than relying on heuristically defined functions. Radlinski et al. (2008a) used a multi-armed bandit model to minimise query abandonment by producing a diverse result list[6]. Xia et al. (2015) built positive and negative rankings for each training query and used a perceptron-based model to optimise diversity evaluation metrics. Several other diversification models have also been suggested in the literature based on structured prediction (Liang et al., 2014b; Yue and Joachims, 2008), Markov decision process (Feng et al., 2018; Xia et al., 2017), and recurrent neural networks (Jiang et al., 2017).

The primary dimension of search results diversification is topical by nature which is the focus of this chapter. However, diversification can be performed from other dimensions. Kacimi and Gamper (2011) studied diversification for controversial queries using the sentiment towards the search topic alongside document relevance and topical diversification. Aktolga and Allan (2013) diversified search results based on documents' sentiment polarity.

[6] Abandonment occurs when a user does not click on any document in the result list.

In their work, diversification could mean: a balanced representation of the different sentiments towards the search topic; a biased list towards the popular sentimental view of the topic or the least popular one. Time represents another dimension according to which results can be diversified. In the context of blog feed retrieval, Keikha et al. (2012) favour blogs that cover an extended time window compared with those published on a similar time. Aktolga (2014) uses content features to detect temporal aspects of the document based on which diversification is performed.

5.3 Evaluation

In chapter 2, I reviewed standard IR evaluation metrics which are used to evaluate search systems that rely on the PRP, i.e. a ranked-list of decreasing relevance to a user query assuming *a single* information need. Diversification as an IR task represents a departure from such a principle. New evaluation metrics and test collections have been developed to accommodate such a change in the underlying assumptions. In this section, I start by reviewing diversity metrics followed by an introduction to diversity test collections.

5.3.1 Evaluation metrics

Subtopic-Recall: Zhai et al. (2003) introduced a simple evaluation metric for the diversity related problem of subtopic retrieval. The basic intuition behind this measure is that a diversified result list should cover as many subtopics of the query as possible. It measures the proportion of a query's subtopics that are covered up to rank k. It is defined as:

$$S - Recall@k = \frac{| \cup_{i=1}^{k} subtopics(d_i)|}{|T|} \tag{5.14}$$

where T is the set of subtopics for the query and $subtopics(d_i)$ is the set of subtopics that is covered by document d_i.

α-nDCG: Clarke et al. (2008) presented α-nDCG as an extension to nDCG to account for redundancy in the result list. Assuming that each query might represent multiple subtopics, a document is relevant to the query if it is relevant to any of its subtopics. Thus, the gain value for a document in α-nDCG settings depends on the number of subtopics that the document covers. This value is further penalised if document k covers subtopics that have already been fulfilled by documents up to rank $k-1$. Formally and assuming a query with m subtopics, the gain value for document k is calculated as follows:

$$G[k] = \sum_{i=1}^{m} J(d_k, i)(1-\alpha)^{r_{i,k-1}} \tag{5.15}$$

where $J(d_k, i)$ is the relevance of document d_k to the query's subtopic i. $r_{i,k-1} = \sum_{n=1}^{k-1} J(d_n, i)$ which is the number of documents relevant to subtopic i up to rank $k-1$. α is the penalisation parameter with values in the range $[0, 1)$. If $\alpha = 0$ then α-nDCG will be equivalent to the standard nDCG. The discounted cumulative gain is then defined as:

$$DCG[k] = \sum_{i=1}^{k} \frac{G[i]}{log_2(i+1)} \tag{5.16}$$

Intent-aware measures: A family of metrics were introduced by Agrawal et al. (2009) to account for differences in the relative importance of a query's subtopics. An example of this is the query *java* where document $d1$ is highly relevant to *the coffee addict* and $d2$ is equally relevant to *the programmer.* A diversification metric that does not factor in the importance, or popularity[7], of these subtopics would treat both ordering $(d1, d2)$ and $(d2, d1)$ equally while in practice $(d2, d1)$ might be a preferable order assuming that the programming subtopic is the most common intent behind this query. Agrawal et al. suggested computing an $NDCG$ score for each query subtopic t by assuming that it is the only relevant subtopic $NDCG(k|t)$. This score is then weighted by the subtopic importance $P(t|q)$. These scores would then be aggregated as follows:

$$NDCG - IA[k] = \sum_{t} P(t|q) NDCG(k|t) \tag{5.17}$$

A similar procedure is followed to compute intent-aware version for Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP) as follows:

$$Metric - IA[k] = \sum_{t} P(t|q) Metric(k|t) \tag{5.18}$$

ERR-IA: In chapter 2, I introduced Expected Reciprocal Rank (ERR) (Chapelle et al., 2009).The diversity measure variant of ERR is computed in a similar way as the intent aware metrics discussed above.

$$ERR[k] = \sum_{r=1}^{k} \frac{1}{r} \sum_{t} P(t|q) \prod_{i=1}^{r-1} (1 - R_i^t) R_r^t \tag{5.19}$$

where R^t maps relevance grade to probability with respect to subtopic t.

D and D♯ measures: Sakai and Song (2011) argue that two properties should be accounted for when evaluating diversified search results. Firstly, retrieved documents should cover as much of the query's subtopics as possible. Secondly, documents that are highly relevant to popular subtopics should be presented before marginally relevant documents to less popular subtopics. The first property is satisfied by the S-Recall measure

[7] The importance of a subtopic is usually estimated based on query logs data.

discussed earlier. The second property entails a graded relevance framework for which they used DCG. They calculated *a global gain* value for each document $GG(d)$, which is essentially a different way of defining a gain value for document d, as follows:

$$GG(d) = \sum_{t \in T} P(t|q)p(rel = 1|t, d) \tag{5.20}$$

In such settings, there would be only *one* ideal gain vector in contrast with an ideal gain vector *per each subtopic* as in the Intent-Aware family. Measures that use the global gain value are prefixed with $D-$. To accomplish the two properties, linearly combining S-Recall and a D-measure is suggested. In the case of D-nDCG, the resultant measure is called D♯-nDCG and is defined as[8]:

$$D\sharp - nDCG = \gamma S - Recall@k + (1 - \gamma)D - nDCG@k \tag{5.21}$$

Novelty Rank-Biased Precision (NRBP): Clarke et al. (2009a) introduced a measure called $NRBP$ that combines features from α-NDCG and the Intent-aware measures within the context of the Rank-Biased Precision (RBP) user model (Moffat and Zobel, 2008). In such a user model, the user proceeds to examine the next document in a result list with probability β and leaves with probability $1 - \beta$. A patient user will have a higher β value. In NRBP settings, a query q may have multiple interpretations S. Each interpretation s can have multiple subtopics T_s. It is defined as follows:

$$NRBP = \frac{1 - (1 - \alpha)\beta}{\sigma} \sum_{k=1}^{\infty} \beta^{k-1} \sum_{n=1}^{S} \frac{p_n}{|T|} \sum_{i=1}^{T_n} J(d_k, n, i)(1 - \alpha)^{r_{n,i,k-1}} \tag{5.22}$$

where p_n is the relative popularity for interpretation n and $J(d_k, n, i)$ is the relevance of document d_k to subtopic i of interpretation n. The ideal score σ normalises the score.

One way to evaluate these metrics is based on their *discriminative power* (Sakai, 2006). This is achieved by measuring the sensitivity of a particular evaluation metric to test search queries. In the case of two IR systems, a robust metric should be able to predict which system is performing consistently better across different sets of test queries (Sakai and Song, 2011). It is computed by running a statistical significance test on every pair of experimental runs. The discriminative power of a metric is the percentage of pairs that are significant at some significance level (Sakai, 2006). Clarke et al. (2011a) have shown that the simple measure of Subtopic-Recall tends to have more discriminative power than those based on a cascade user model such as α-nDCG and NRBP. Sakai and Song (2011) validated evaluation metrics when per-intent graded relevance is used.

[8] The effect of γ is not significant as both measures are highly correlated (Sakai and Song, 2011). The default value is $\gamma = 0.5$.

They found D♯ measures, α-nDCG and Subtopic-Recall to be the most discriminative measures for shallow depth evaluation. Another method of validating metrics is based on their *predictive power*. Sanderson et al. (2010) studied the correlation between evaluation metrics and users' preferences and found that diversity metrics correlated reasonably well with users' preferences with Subtopic-Recall being as effective as α-nDCG and NRBP. There are other studies that compare diversity measures based on their informativeness (Ashkan and Clarke, 2011) or using rank correlation measures (Clarke et al., 2011a). In general, it is considered a good evaluation practice to report a variety of measures when evaluating diversification systems (Clarke et al., 2011a) since some of these measures might be assessing different properties of the search results (Sanderson et al., 2010).

5.3.2 Test collections

Two evaluation conferences have produced test collections suitable for evaluating diversification approaches. Between 2009 and 2012, TREC ran a diversity task for the specific purpose of evaluating search results diversification approaches (Clarke et al., 2009b, 2010, 2011b, 2012). This effort produced four test collections. NTCIR have also built test collections for diversification for Asian languages as well as the English language through their Intent and IMine tasks (Liu et al., 2014; Song et al., 2011). Throughout this chapter, I use TREC test collections to evaluate my approach.

The diversity tasks of TREC 2009, 2010, 2011, and 2012 used ClueWeb09[9] as their document collection, which is a collection of web documents crawled in early 2009. It is distributed into two categories. The first category contains 1.04 billion web pages in 10 languages with about 500 million pages in English (category A). The second category, referred to as ClueWeb09 category B or ClueWeb09B, is a subset of 50 million English web documents. This set is referred to as ClueWeb09 category B (ClueWeb09B). Participants of TREC diversity tasks used both categories.

Each task consists of 50 topics which were developed by NIST (National Institute of Standards and Technology) using queries sampled from the logs of a commercial search engine. For TREC 2009 and 2010, the sampling process favoured queries with medium popularity while in TREC 2011 and 2012 less popular queries were preferred. The diversity task handles two types of query's ambiguity. The first type is *ambiguous* queries referring to queries with multiple interpretations. The second is *faceted* queries for queries with a clear interpretation but multiple facets. Figure 5.1 presents a sample extracted from TREC 2009 diversity task. As shown in the figure, each topic consists of multiple subtopics.

To reflect real users' information needs, a clustering approach, which uses query reformulation and co-clicks data, was used to help the NIST team to infer each query's subtopics (Radlinski et al., 2010). Documents were judged based on their relevance to

[9] http://lemurproject.org/clueweb09/.

```
<topic number="5" type="faceted"><query>mitchell college</query>
<description> Find information about Mitchell College in New London,
CT, such as a prospective student might find useful. </description>
<subtopic number="1" type="nav">
Find the homepage for Mitchell College.</subtopic>
<subtopic number="2" type="nav">
Find the homepage for the athletics department at Mitchell College.
</subtopic><subtopic number="3" type="inf">
Find web pages that compare Mitchell College to other colleges in
Connecticut.</subtopic>
<subtopic number="4" type="inf">
Find information on admissions to Mitchell College. How do I
become a student there? </subtopic></topic>
...
<topic number="25" type="ambiguous"><query>euclid</query>
<description>Find information on the Greek mathematician Euclid.
</description><subtopic number="1" type="inf">
Find information on the Greek mathematician Euclid.</subtopic>
<subtopic number="2" type="inf">
I’m looking for a source for Euclid truck parts.</subtopic>
<subtopic number="3" type="nav">
Take me to the homepage for Euclid Industries.</subtopic>
<subtopic number="4" type="nav">
Take me to the homepage for the Euclid Chemical company.
</subtopic></topic>
```

Figure 5.1: An example from TREC 2009 web diversity track. Topic number 5 has one plausible interpretation (The Mitchell College in New London, CT) but is underspecified. Topic number 25 is an example of a search query that has multiple interpretations.

each subtopic using binary relevance[10] as well as their relevance to the topic as a whole. Table 5.1 presents statistics of the evaluation datasets used in this chapter.

5.4 Diversification based on social positions

In section 5.2, I reviewed various methods to diversify search results. Most of these methods require a set of subtopics for each test query. A subtopic can be a facet for multi-faceted queries or an interpretation, a sense, for an ambiguous query. There are two main approaches to handling this requirement: explicit and implicit. An explicit approach extracts subtopics from an external resource such as query logs or query recommendation services or even the gold standard subtopics provided by diversification tasks’ organisers

[10] TREC 2011 and 2012 use binary relevance but make graded relevance available. NTCIR differentiates itself from TREC by using graded relevance and 100 test queries instead of 50.

	2009	2010	2011	2012
# queries with relevant documents.	50	48	49	50
# ambiguous queries.	12	27	9	10
# faceted queries.	38	23	41	40
Subtopics per query (average).	4.86	4.36	3.36	3.90
Relevant documents per query (average).	81.62	115.04	91.65	121.72

Table 5.1: Summary statistics of TREC diversification datasets. These figures were calculated on ClueWeb09B.

(Dang and Croft, 2012; Santos et al., 2010). The focus of such methods is on diversification strategies more than the sub-task of subtopics identification or representation. The implicit approach relies mostly on clustering (Carterette and Chandar, 2009) or summarisation (Dang and Croft, 2013) techniques to provide a surrogate representation of each query's subtopics. The goal of this chapter is to investigate various implicit representations and diversification strategies for cases where external resources are not available to extract subtopics directly.

In this section, I introduce social positions as an implicit approach to represent query's subtopics for the task of search results diversification. Each social position is an interpretation, a view, of the query that can be composed of multiple subtopics. The proposed approach consists of two steps. The first is presented in section 5.4.1. The purpose of this step is to identify social positions of each test query. The second step, presented in section 5.4.2, is concerned with the diversification strategy that takes a query's social positions into account. In section 5.5, I present the experimental setting, which includes evaluation methodology and baselines used to validate the proposed approach. Results are discussed in section 5.6 while section 5.7 provides a chapter summary.

5.4.1 Matching search queries to social positions

As mentioned earlier, the search query itself is often insufficient as the only representation of a user's information need. Previous research in areas such as sense induction (Navigli and Crisafulli, 2010) and query expansion (Carpineto and Romano, 2012), among others, have considered documents returned for a search query as a source of an additional context to be used to interpret or expand the query. In this section, I assume that the top k ranked documents for a query are representative of the query's candidate social positions. For example, the search query *java* is likely to return documents relevant to *a java programmer*, *a coffee addict,* or *a tourist.* Thus, I formulate this task as a document classification task. The goal is to assign a search query to one or more social positions using the top k ranked documents for each query as a representation of the query.

There are two research questions to be solved with respect to the above goal. The first is how to estimate the degree by which a document d belongs to a particular social position r. The second question is whether *all* documents returned for query q should be used to represent it or *only some* of them. Supervised machine learning approaches are often adopted to solve the first question. Unfortunately, this requires a costly development of a training corpus. Instead, I apply a language modelling approach using the models I built for each social position as described in chapter 4. The DiffLDA model learns a multinomial distribution over words for each social position. Each social position is then represented by the top n words in its probability distribution. The goal is to estimate $P(d|r)$ by treating the multinomial distribution produced by the DiffLDA model for each social position as a weighted query. $P(d|r)$ would then be equal to $P(d|q)$ and can be estimated using the QL model or any other probabilistic retrieval model.

Upon manual examination, my initial results using this technique were of low quality. There are probably two reasons that could explain this behaviour. Firstly, not all terms that represent a social position are indeed relevant to the social position since it is typical for LDA-based topics that are estimated using noisy sources such as web documents to contain extraneous terms (Newman et al., 2011). Secondly, the QL model, as with most standard retrieval models, assumes independence between query terms. In contrast, a social position is made up of a group of terms that would tend to co-occur with each other. Following the independence assumption, a few words frequently occurring in a single document will boost the likelihood score for the social position they belong to. This is especially problematic for common terms that might have been mistakenly assigned to a social position by the DiffLDA model.

I, therefore, introduce a distance measure $S(d, r)$ that can be used to calculate the similarity between a web document d and a social position r. Each social position is represented by n documents. These documents are the top $n = 10$ documents retrieved from the ClueWeb09 collection using the social position model as a weighted query.

The distance measure $S(d, r)$ is then defined as the average cosine distance between a web document d and $\mathcal{D}_r$ which is the n documents that represents the social position r:

$$S(d, r) = \frac{1}{n} \sum_{\hat{d} \in \mathcal{D}_r} cos(\hat{d}, d) \tag{5.23}$$

A Direct implementation of equation 5.23 would require $10 \times R \times K$ similarity calculations where R is the number of social positions and K is the number of documents. Most of these calculations may not be necessary, especially if query matching is to be done online. Hence, I develop a pruning strategy to eliminate unlikely social positions. Firstly, I cluster the set of documents D_q for the search query q into $T = 50$ topics using a standard LDA implementation. The aim of this is to divide each document into segments of topically related terms. Each segment can then be scored using each social position model as a weighted query and the QL function as the retrieval model. This segmentation process helps in lessening the effects of noisy terms in social position models and the retrieval model's independence assumption that were found in my initial matching experiment. I extract 10 candidate social positions per segment per document and then rank candidate social positions by frequency and use the top 100 as the set of candidate social positions R_q for query q. The number of segments per document is set to 3 because in the LDA model, which is used to cluster documents, the α hyper-parameter is set to the typical value of $50/T$ (Griffiths and Steyvers, 2004). This value generally means few topics will be used to generate each document. A document d is assigned to social position r if the distance score is above a pre-specified threshold θ. Since documents below the specified threshold would not be assigned a social position, some queries may not also be assigned a position. For example, $\theta = 0.50$ would result in not assigning any social position to the majority of queries. This is clearly undesirable. I experimented with various values of θ and set $\theta = 0.10$, which would result in about 11% of queries not assigned to any social position.

I now address the second question of whether *all* documents returned for a query should be used to represent it or *only some*. The definition of *all* is documents up to a rank cutoff $k = 200$ from the initial retrieval list. The inclusion of all documents in the representation of a search query places a high level of trust in the initial retrieval model. In other words, we are assuming that all documents returned for a search query are relevant. This assumption is analogous to the pseudo-relevance assumption, although the rank cutoff in the pseudo-relevance feedback is much lower. For example, consider the query *Obama family tree* from TREC web track 2009. If we consider all documents as a representation, the two dominant social positions representing this query are: genealogist (47.5%) and Obama supporter (26%). It might be clear that the terms that triggered these two social positions are *family tree* and *Obama*, respectively. However, the main concept or entity in this query is actually *Obama* and it is intuitive to assume that a relevant document for this query must contain the term *Obama*. Table 5.2 shows the top 10 documents retrieved using the QL model for this query and the distance measure score (equation 5.23) for each document with respect to *a genealogist* and *an Obama supporter*. As shown in the table, the social position *genealogist* appears to be assigned to 9 of the top documents while 8 of those documents do not contain the term *Obama* which is the key concept of the query. Again, this is probably caused by the initial ranker's ignorance of term dependency and weighting all query terms equally. These issues would result in a misleading distribution of social positions (47.5% genealogist and 26% Obama supporter).

These challenges have long been recognised and several dependency models (e.g.

	Genealogist	**Obama supporter**	**Contains *Obama***	**Relevant**
clueweb09-en0009-30-02857	0.23	0.01	✗	✗
clueweb09-en0007-63-02101	0.06	0.01	✗	✗
clueweb09-en0009-30-02446	0.23	0.01	✗	✗
clueweb09-en0009-30-02807	0.23	0.01	✗	✗
clueweb09-en0009-30-02747	0.23	0.01	✗	✗
clueweb09-en0009-30-02678	0.23	0.01	✗	✗
clueweb09-en0001-02-21241	0.13	0.40	✓	✓
clueweb09-en0009-30-02922	0.24	0.01	✗	✗
clueweb09-en0009-30-02621	0.23	0.01	✗	✗
clueweb09-en0009-30-02795	0.24	0.01	✗	✗

Table 5.2: Example social positions assignments for topic #1 *Obama family tree* in TREC web track 2009. The relevance column is based on the relevance assessment provided by the task organisers.

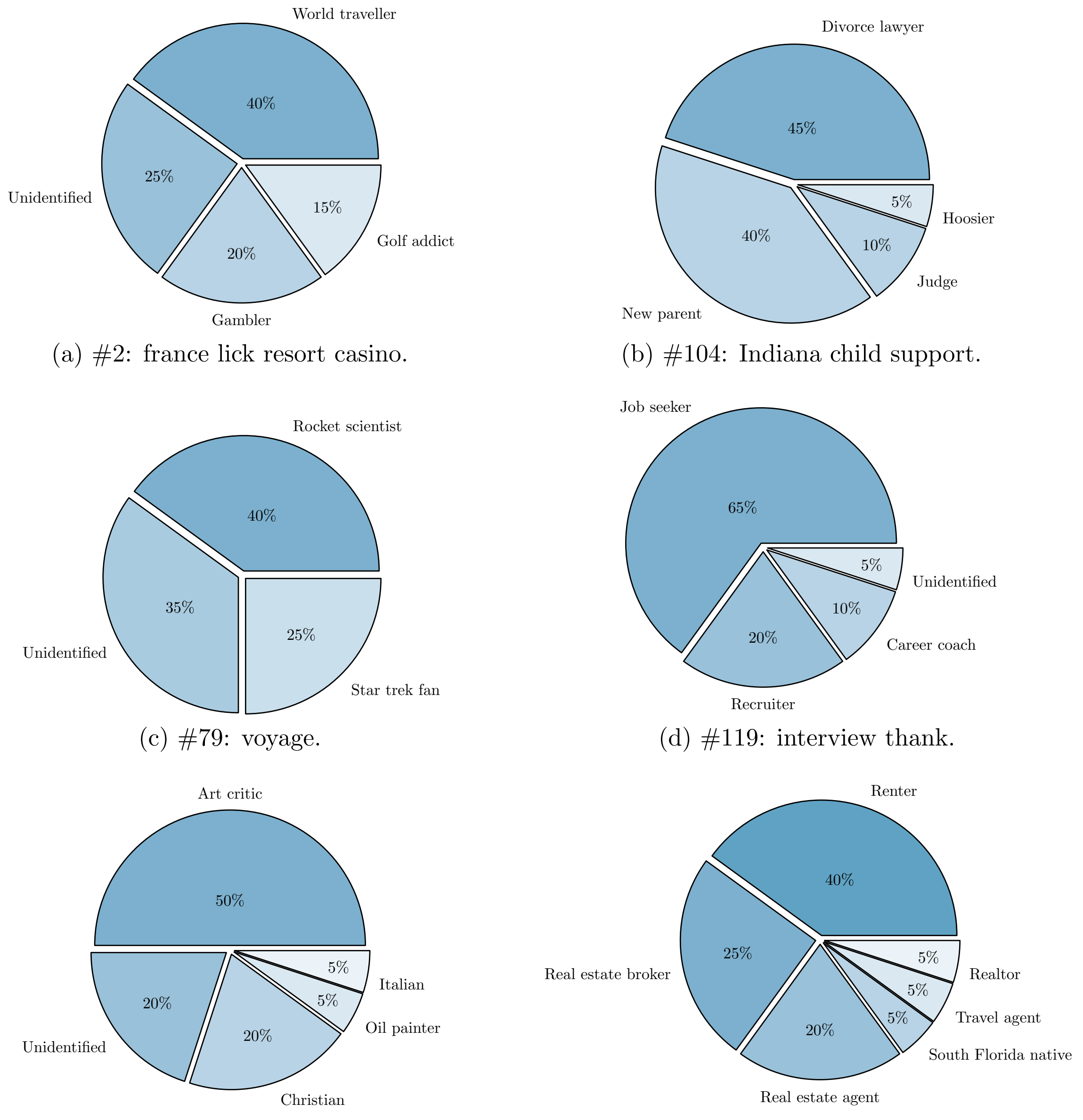
Bendersky et al., 2010; Metzler and Croft, 2005; Peng et al., 2007) and key concept weighting methods (e.g. Bendersky and Croft, 2008; Zhao and Callan, 2010; Zheng and Callan, 2015) exist. The goal, however, at this stage is not to improve the retrieval performance of the initial ranker but to select specific documents from the initial list that are likely to be relevant to the query. These documents would be used to estimate the distribution of social positions in a two-stage process. First, segmenting the query in order to identify the query's key concept and then to select documents based on the identified key concept.

I use simple heuristics to segment search queries. Firstly, I assume each query contains a single key phrase, i.e. bigram[11], where each bigram in the original query is a candidate. Secondly, candidates are ranked based on their frequency of occurrences in the initial result list. Finally, the top candidate is considered the key phrase of the query if it occurs at least *200 times*. If the query does not have a key concept, the most frequent term[12] would be the key concept.

In the second stage, a document is chosen to represent the query if a snippet from the document can be extracted. A snippet must contain the query's key phrase and at least one of the other query terms within a window of 10 words. Based on this document selection procedure, the social positions distribution for the example query *Obama family tree* shifts considerably in favour for *Obama supporter* (from 26% to 95%) compared with *Genealogist* (from 47.5% to 5%).

[11] Exact phrase matching using query bigrams has been shown to improve retrieval performance (Bendersky et al., 2011c).

[12] Queries are pre-processed to remove stopwords and numbers.



(a) #2: france lick resort casino.

(b) #104: Indiana child support.

(c) #79: voyage.

(d) #119: interview thank.

(e) #155: last supper painting.

(f) #192: condo florida.

Figure 5.2: Distributions of assigned social positions for example queries.

5.4.2 Diversification strategy

The output of the previous section is a social positions distribution for each query. Examples are shown in figure 5.2. Formally, let us have $P(r|q)$ which is the probability that query q belongs to social position r. $P(r|q)$ is then calculated as follows:

$$P(r|q) = \frac{\sum_{d \in D_q} \Phi(d, r)}{|D_q|} \tag{5.24}$$

where D_q is the set of documents that were selected to represent query q and $\Phi(d, r)$ is

Position: Art critic	
1	image, art, famous, code, artist, paint, history, supper, picture, oil, painting, style, detail
2	christ, london, artist, collection, medium, house, oil, watercolor, acrylic, reproduction
3	fresco, original, artist, work, judas, painting, apostle, john, restoration, table
Find a picture of the Last Supper painting by Leonardo da Vinci.	
Position: Christian	
1	bible, christ, crucifixion, cross, artwork, life, jesus, oil, christian, lamb, god
2	caravaggio, work, judas, supper, version, vinci, jesus, painting, apostle, john
3	religious, bread, christ, passover, church, disciple, meal, century, jesus, reformation, christian, gospel, john, wine
What is the significance of da Vinci's interpretation of the Last Supper in Catholicism?	
Position: Italian	
1	religious, church, renaissance, venice, world, today, things, know, time, italy, story
2	image, ticket, city, alto, milan, feature, geography, visit, travel, map, italy, guide
3	art, africa, christ, crucifixion, saint, middle, metropolitan, medieval, scene, america, century, museum, timeline, europe, italy
Are tickets available online to view da Vinci's Last Supper in Milan, Italy?	

Table 5.3: Example of subtopics for the query *last supper painting* and its matching to the gold-standard subtopics provided by TREC.

a function defined as follows:

$$\Phi(d,r) = \begin{cases} 1 & \text{if } S(d,r) \geqslant 0.10 \\ 0 & \text{Otherwise.} \end{cases}$$

Note $P(r|q)$ is not the same as $P(t|q)$ that is prevalently used in most diversification algorithms, as reviewed in section 5.2. $P(t|q)$ is the probability that query q is about subtopic t. The difference is that a subtopic t is usually a fine-grained representation of a probable information need. In contrast, a social position r is a coarse-grained representation. There are multiple possible subtopics for each social position.

For example, consider the query *last supper painting* in figure 5.2e. The user can be *an art critic (*50%*), a Christian (*20%*), an oil painter (*5%*),* or *an Italian (*5%*).* For the art critic user, several subtopics could be of interest such as *the painting medium, its restoration, or reproduction.* Table 5.3 shows an example of subtopics for this query and its tentative matching to the gold-standard subtopics provided by TREC.

I construct a diversified results list for each social position that has a $P(r|q) > 0$ for each query. Formally, let us have $D_{q,r}$ as the set of documents for query q that are assigned to social position r using equation 5.23. I cluster $D_{q,r}$ into N topics using a standard LDA model in order to discover the subtopics for query q from the perspective of social position r. Then, I diversify $D_{q,r}$ using a diversification function such as IA-Select or xQuAD. This results in $S_{q,r}$ which is a diversified list of documents for query q with respect to social position r. The final diversified list F_q for query q is built by firstly selecting the most probable social position r^*:

$$r^* \leftarrow \arg\max_{r \in R_q} P(r|q) \tag{5.25}$$

Then, the first document d^* in S_{q,r^*} is added to F_q:

$$\begin{aligned} d^* &\leftarrow \text{pop } S_{q,r^*} \\ F_q &\leftarrow F_q \cup \{d^*\} \end{aligned} \tag{5.26}$$

I aim to proportionally divide the final diversified list F_q between the query's social positions R_q based on the probability of each social position $P(r|q)$. This is similar to the PM-1 and PM-2 approaches suggested by Dang and Croft (2012) apart from two aspects. Firstly, the list of documents for each social position is a diversified list rather than a list of documents ranked by their relevance score as in PM-1. Secondly, my approach and the PM-2 deal with the fact that a document can be relevant to more than one topic or social position but the quotient, i.e. $P(t|q)$, or $P(r|q)$ in my approach is updated differently. PM-2 decreases $P(t|q)$ for all topics proportionally to their normalised relevance to the selected document while I update $P(r|q)$ by subtracting a constant value equal to $\frac{1}{K}$ where K is the size of the final diversified list as in equation 5.27 ($K = 20$ in all TREC diversity tasks).

$$\forall r \in R_q, P(r|q) = P(r|q) - \frac{1}{K} \iff S(d^*, r) \geqslant 0.10 \tag{5.27}$$

5.5 Experimental setup

The main research questions of this chapter are:

RQ1. How competitive is diversification based on social positions compared with other implicit approaches?

RQ2. What factors are influencing the performance of diversification based on social positions?

RQ3. Does diversification based on social positions favour particular types of queries over others?

In this section, I describe the details of my experimental setup adopted to answer the above questions. In section 5.6, I report my results.

5.5.1 Test queries and retrieval collection

I use 200 test queries that were developed by the diversity task of TREC web track. The task ran for four consecutive years from 2009 with a set of 50 test queries used each year (Clarke et al., 2009b, 2010, 2011b, 2012). Details about this test collection are provided in section 5.3.2. All my experiments are performed using ClueWeb09B as the document collection and an experimental search engine I developed. Documents and test queries were stemmed using the Krovetz stemmer (Krovetz, 1993) and stopwords were removed[13].

5.5.2 Evaluation procedure

The diversity task used a variant of intent-aware expected reciprocal rank (*ERR-IA*), reviewed in section 5.3.1, as the primary evaluation metric from 2010 (Clarke et al., 2010, 2011b, 2012). I also used *ERR-IA* as the primary metric to evaluate diversification approaches. It is also standard to report other diversity measures such as α-*NDCG*, *NRBP* and *S-Recall*. All runs were evaluated using the official relevance judgments and evaluation code provided by task organisers. I performed a paired t-test with Bonferroni correction to examine the statistical significance of test runs ($p < 0.05$).

All of my experiments were based on a re-ranking approach in which an initial list of results for each query is processed to produce the final list. I first retrieved the initial list using the QL model (Ponte and Croft, 1998) with a Dirichlet smoothing parameter (μ) set to 3500. I, then, removed documents that are likely to be spam documents. The web contains spam pages that are designed to trick search engines into ranking them higher than legitimate web pages. Cormack et al. (2011) have shown that filtering out spam pages results in significant improvements in retrieval performance. A number of IR studies have since then applied such a filtering process (e.g. Dang and Croft, 2013, 2012; Guan et al., 2013; He et al., 2012; Luo et al., 2014b; Zamani and Croft, 2017). This was done using the technique suggested by Cormack et al. (2011), which assigns a percentile for each document. A low percentile suggests that all documents that fall beneath it are likely to be spam documents. I removed documents with a percentile < 70. Lastly, this initial list was cut at rank 200. I refer to this rank cutoff parameter throughout this chapter as k. Algorithm 4 summarises the diversification process followed in my experiments.

[13] I use Indri's stopwords list available at www.lemurproject.org/stopwords/stoplist.dft

Algorithm 4: A re-ranking greedy approach to search results diversification.

```
Input: A search query $q$, a set of subtopics or aspects $T_q$, an initial set of
       documents $S_q$ and a cutoff rank $k$
Output: A diversified list of documents $R_q$
begin
    $R_q \longleftarrow \emptyset$ ;
    while $|R_q| < k$ do
        $d^* \longleftarrow \arg\max_{d \in S_q} f(q, T_q, S_q, R_q)$ ;
        $R_q \longleftarrow R_q \cup \{d^*\}$ ;
        $S_q \longleftarrow S_q \setminus \{d^*\}$ ;
    end
    Return $R_q$ ;
end
```

5.5.3 Baselines

Most diversification algorithms require a subtopic representation per query. These subtopics can be explicitly extracted from sources such as query logs or commercial search engines' services (Santos et al., 2010). They can also be implicitly modelled using the initial retrieval list of documents for each query. Comparisons are made with implicit methods. In this section, I describe four different implicit methods which I used as baselines to model query's subtopics. These methods should provide a set of subtopics $T(q)$ per query. They must also provide two probabilistic estimates. First, $P(t|q)$ which is the probability that subtopic t is relevant to query q. Second, $P(d|t)$ which is the probability that document d is generated by subtopic t. Figure 5.3 presents a summary of diversification algorithms used in my experiments. These have been reviewed in sections 5.2.1 and 5.2.2.

LDA: Blei et al. (2003) introduced LDA as a generative probabilistic model. It has since then accumulated widespread recognition as an effective unsupervised topic model. LDA assumes that a document collection is made up of a number of topics, which in turn are used to generate each document. Each topic is a multinomial distribution over words. This fits naturally with the problem of query's subtopics identification. By applying LDA to D_q, which is the set of documents to be diversified for query q, we can treat each topic that is estimated by LDA as a subtopic for query q. Both values $P(t|q)$ and $P(d|t)$ come freely in the process of estimating the LDA model. The former represents the extent to which topic t contributes to generating the entire collection D_q while the latter is equal to the smoothed probability of document d being generated from topic t. As discussed in chapter 4, the Gibbs sampling algorithm assigns each word w to a specific topic z. Using these estimates, the topic-document distribution θ_d for document d can be obtained as follows:

$$\theta_d^t = \frac{\Omega_{d,t} + \alpha}{\sum_{z=1}^{T} \Omega_{d,z} + T\alpha} \tag{5.28}$$

$$d^* \leftarrow \arg\max_{d \in S_q} \Big(\lambda \times P(d|q) \Big) - \Big((1-\lambda) \times \arg\max_{\hat{d} \in R_q} F_{sim}(d, \hat{d}) \Big)$$

MMR (Carbonell and Goldstein, 1998)

$$d^* \leftarrow \arg\max_{d \in S_q} \Big(\lambda \times P(d|q) \Big) + \Big((1-\lambda) \times \sum_{t \in T_q} P(t|q) \times P(d|t) \times \prod_{\hat{d} \in R_q} 1 - P(\hat{d}|t) \Big)$$

xQuAD (Santos et al., 2010)

$$d^* \leftarrow \arg\max_{d \in S_q} \Big(\lambda \times P(d|q) \Big) + \Big((1-\lambda) \times \sum_{t \in T_q} P(t|q) \times P(d|t) \times (2 - 2 \sum_{\hat{d} \in R_q} P(\hat{d}|t) - P(d|t)) \Big)$$

Square loss (Zheng et al., 2012)

$$d^* \leftarrow \arg\max_{d \in S_q} \Big(\lambda \times quotient[t^*] \times P(d|t^*) \Big) + \Big((1-\lambda) \times \sum_{t \neq t^*} quotient[t] \times P(d|t) \Big)$$

PM-2 (Dang and Croft, 2012)

$$d^* \leftarrow \arg\max_{d \in S_q} \sum_{t \in T_q} U(t|q, R_q) \times \Big(P(d|t) \times P(d|q) \Big)$$

IA-Select (Agrawal et al., 2009)

$$\forall t \in T(d^*), U(t|q, R_q) = \Big(1 - \Big(P(d^*|t) \times P(d^*|q) \Big) \Big) \times U(t|q, R_q \setminus \{d^*\})$$

where $U(t|q, R_q)$ is initialised $\forall t$ as: $U(t|q, R_q) = P(t|q)$

Figure 5.3: A summary of diversification functions used in this chapter. These functions replace line 4 in algorithm 4.

$\Omega_{d,t}$ is the number of times topic t is assigned to document d and T is the number of topics. In diversification algorithms, $P(d|t)$ is equivalent to θ_d^t. To estimate $P(t|q)$, all documents for query q up to rank k are concatenated into a single document. $P(t|q)$ is then calculated in a similar way as $P(d|t)$. I use my own LDA implementation and optimise LDA hyper-parameters using the MALLET package (McCallum, 2002). A detailed review of LDA is provided in chapter 4.

K-means: Another possible method to obtaining subtopics is to cluster the initial set of documents D_q. Each cluster can then be considered a subtopic. In this baseline, I use the k-means as the clustering algorithm. K-means is essentially a hard clustering technique in which a document is assigned to one cluster only. I use this property to estimate the importance of cluster t to query q as follows:

$$P(t|q) = \frac{|D_{q,t}|}{|D_q|} \tag{5.29}$$

where $D_{q,t}$ is the set of documents assigned to cluster t of query q. To estimate $P(d|t)$, I follow a similar process used by Carterette and Chandar (2009) and Dang and Croft (2013). I built a relevance model, truncated at rank 20, for each cluster using the RM1 method of Lavrenko and Croft (2001). This relevance model is treated as a weighted query and the relevance of each document d to this weighted query is estimated using the QL model to produce $P(d|t)$. Note that this process removes the hard constraint as each document will have a smoothed $P(d|t)$ for all subtopics. I linearly transform $P(d|t)$ to the range $[0.25, 0.75]$.

Spectral clustering: In this baseline, I use a spectral clustering method (Ng et al., 2001) as the subtopic identification algorithm. One intuitive explanation of spectral clustering is based on the concept of a random walk. Assuming a weighted graph, spectral clustering aims to let a random walker stochastically jump within a cluster and rarely move to another cluster. Similarly, a user who is interested in one interpretation of a query is more likely to stay within that same interpretation for the duration of a search session. I construct a fully connected graph for each D_q using the cosine similarity to weight the graph edges. I follow the same process to estimate $P(t|q)$ and $P(d|t)$ as I did with the k-means baseline since spectral clustering is also a hard clustering technique[14].

DSPApprox: Dang and Croft (2013) suggested using the hierarchical summarisation algorithm (DSPApprox) of Lawrie and Croft (2003) as a method of finding query's subtopics. DSPApprox depends on two probabilistic notions: term's topicality and predictiveness. Topicality is a measure of a term's relevance to a given topic while predictive terms are those that predict the occurrence of other vocabulary terms. The goal of DSPApprox is to select a set of terms that maximises the joint probability of both concepts. The topicality of term t is estimated as follows:

$$KL\ contribution(t) = P(t|q) log_2 \frac{P(t|q)}{P_c(t)} \tag{5.30}$$

where $P(t|q)$ is calculated using a relevance model (Lavrenko and Croft, 2001) and $P_c(t)$ is a language model of ClueWeb09B. Topical terms are expected to have a positive KL contribution. To estimate the predictiveness of a term, a co-occurrence language model for terms within a fixed window w is built using frequency estimation. The predictability of term t is then calculated as follows:

$$Predictability(t) = \frac{1}{|V|} \sum_{v \in V} P_w(t|v) \tag{5.31}$$

V is the set of vocabulary terms selected based on similar criteria as Dang and Croft (2013). A term must occur in two documents, not be numeric, and have at least two

[14] I use the Smile Implementation of spectral clustering at http://haifengl.github.io/smile/

characters. DSPApprox aims to select a diversified set of terms by first selecting the term t^* that maximise the joint probability and then decreasing the predictiveness of all topic terms that predict a vocabulary term already covered by the selected term t^*. Topic terms are those that co-occur with any of the query's terms within a fixed window. Similar to Dang and Croft (2013), I set $P(t|q)$ uniformly for all topic terms and calculate $P(d|t)$ as follows[15]:

$$P(d|t) = \Big(P(t|d)P(q|d)\Big)^{\frac{1}{|q|+1}} \tag{5.32}$$

5.5.4 Parameter settings

There are a number of parameters involved in the various systems used in my evaluation. The first is the λ parameter that controls the trade-off between relevance and novelty in diversification functions. I considered values for $\lambda \in [0.05, \ldots, 0.95]$. The other key parameter is the number of subtopics $|T_q| \in \{2, 3, 4, 5, 10\}$ which is used in the identification of subtopics within each social position. It is also used in the clustering baselines: LDA, k-means, and spectral clustering. The LDA implementation was run for 1000 iterations using the suggested hyper-parameter values in Griffiths and Steyvers (2004). For the DSPApprox approach, I considered the following values: $\{2, 3, 4, 5, 10, 20, \ldots, 100\}$ for the number of selected topic terms per query. I set the window size w parameter to the value of 20 as suggested by Dang and Croft (2013). Diversification based on social positions requires a list of documents to be built for each candidate social position. I selected values for the size of this list from $\{30, 40, 50\}$. All of the above parameters are tuned using a four-fold cross-validation setting. When a document has a distance score, equation 5.23, for all candidate social positions < 0.10, the social position of this document was tagged as *unidentified.* Some queries have more than half of their documents labelled as such. In this case, I diversified the top 50 documents of the query regardless of their assigned social positions.

5.6 Results

In section 5.5, I presented the three main research questions of this chapter. These are: (RQ1) how competitive is diversification based on social positions compared with other implicit approaches? (RQ2) What factors are influencing the performance of diversification based on social positions? (RQ3) does diversification based on social positions favour particular types of queries over others? I address question RQ1 in section 5.6.1. RQ2 and RQ3 are discussed in sections 5.6.2 and 5.6.3, respectively.

[15] I implemented DSPApprox and also used the implementation available at (https://github.com/ashishiiith/Adobe-Project) to verify results.

	ERR-IA	α-NDCG	NRBP	S-Recall	+	-	=
QL	0.2798	0.3741	0.2433	0.5769			
MMR	0.2795	0.3740	0.2429	0.5778	18	23	159
LDA							
xQuAD	0.2856	0.3783	0.2497	0.5758	106	59	35
Square loss	0.2774	0.3719	0.2405	0.5645	103	70	27
PM-2	0.2721	0.3660	0.2338	0.5800	87	88	25
IA-Select	0.2835	0.3682	0.2546	0.5488	87	89	24
K-Means							
xQuAD	0.2769	0.3700	0.2410	0.5702	116	53	31
Square loss	0.2834	0.3738	0.2495	0.5632	110	64	26
PM-2	0.2819	0.3765	0.2470	0.5703	109	61	30
IA-Select	0.2820	0.3770	0.2466	0.5752	110	59	31
Spectral clustering							
xQuAD	0.2765	0.3700	0.2407	0.5702	115	54	31
Square loss	0.2827	0.3745	0.2485	0.5634	105	66	29
PM-2	0.2864	0.3813	0.2512	0.5708	109	59	32
IA-Select	0.2853	0.3795	0.2498	0.5717	112	57	31
DSPApprox							
xQuAD	0.2698	0.3646	0.2328	0.5702	114	55	31
Square loss	0.2696	0.3644	0.2326	0.5702	113	56	31
PM-2	0.2614	0.3561	0.2242	0.5643	94	72	34
IA-Select	0.2646	0.3599	0.2275	0.5699	97	66	37
Social positions							
xQuAD (SP)	**0.3147**$^{\mathrm{Q,L}}_{\mathrm{K}}$	**0.4041**$^{\mathrm{Q,L}}_{\mathrm{K}}$	0.2818^{Q,L}	**0.5827**	98	61	41
Square loss	0.2984	0.3797	0.2717	0.5303	85	86	29
PM-2	0.3016	0.3882	0.2704	0.5648	91	82	27
IA-Select	0.3123	0.3984	**0.2822**	0.5781	97	76	27

Table 5.4: Performance of diversification based on social positions compared with other baselines on 200 test queries.

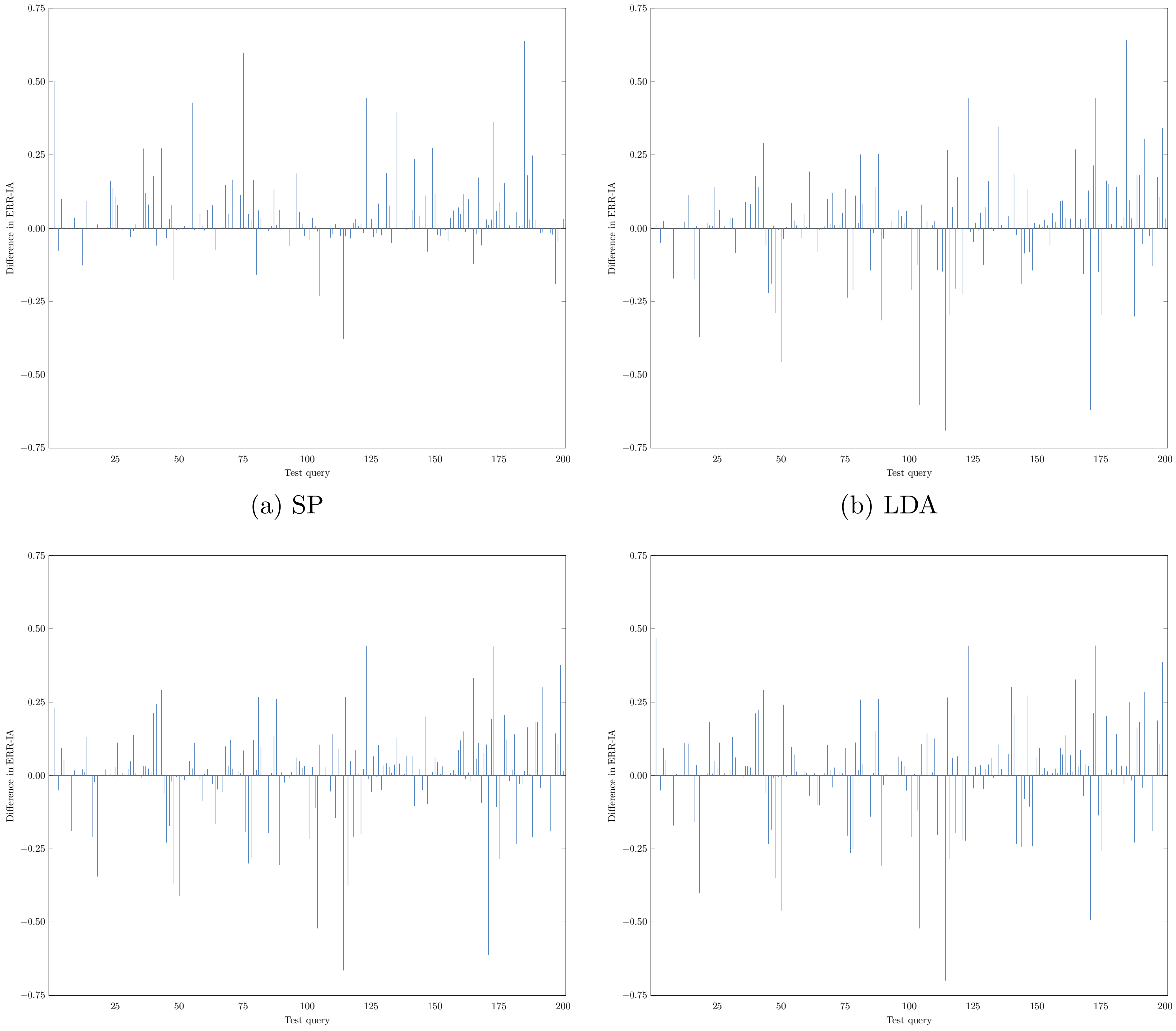
5.6.1 Effectiveness

In this section, I answer RQ1 regarding the competitiveness of social positions as a topic representation for diversification algorithms by comparing it to the baseline subtopics identification methods described in section 5.5.3. Diversification based on social positions uses a function internally to diversify documents relevant to each position per query. I report the results obtained using the four diversification functions: xQuAD, square loss, PM-2, and IA-Select, as in figure 5.3. The same functions are also used in combination with LDA, k-means, spectral clustering, and DSPApprox to produce baseline runs. The importance of subtopic t to query q, $p(t|q)$ is estimated using methods described in section 5.5.3, a non-uniform $p(t|q)$. This does not apply to DSPApprox, which sets a uniform weight for subtopics. I have experimented with setting $p(t|q)$ uniformly for the other approaches which resulted in lower performance, especially when using LDA as topic representation. Table 5.4 presents the results of diversification based on social positions compared with various baselines as well as the QL retrieval model. The best performing run, based on the ERR-IA metric, for each subtopic identification method is underlined. I use letters to indicate statistically significant differences. Q refers to the QL model while L, K, and S refer to the best performing baseline using LDA, k-means, and spectral clustering, respectively.

Three main observations can be made from table 5.4. Firstly, the results indicate that the use of social positions as a method of subtopic identification provides a statistically significant improvement over some competitive and widely used implicit subtopic identification methods. The best performing internal diversification function for social positions was xQuAD which showed statistically significant improvement over three baselines under evaluation metrics: ERR-IA, and $\alpha-$NDCG. The relative improvements over the baseline QL are (+12.5%), (+8%), and (+15%) under ERR-IA, $\alpha-$NDCG, and NRBP, respectively. The IA-Select approach also provided comparable performance to xQuAD. All approaches, including social positions' runs, re-rank the initial list that is obtained using the QL method. Therefore, the subtopic recall for all of these approaches seems to be comparable to that of the baseline run QL.

Secondly, the performance improvement of LDA, k-means, and spectral clustering as implicit methods of subtopics identification is marginal compared with the baseline run QL. Whilst improvements over QL can be seen in some baseline runs, the lack of statistical significance might suggest that these subtopics identification methods failed to provide effective representation of the subtopics behind search queries. The MMR diversification approach, which does not use any subtopic representation, also did not show any improvement over the QL baseline. The small difference between QL and MMR was due to the parameter λ. This parameter controls the trade-off between promoting relevant documents or novel ones. During the parameter settings phase, λ is best at 0.90 for MMR, which means that relevance is heavily weighted and thus the results of MMR will not be of much difference from QL. A similar behaviour is also encountered with the DSPApprox.

Thirdly, the robustness of the SP approach was comparable to the other baselines in terms of the number of affected queries. SP helps 49%, hurts 31%, and keeps 20% of queries unchanged compared with the QL method. To investigate the robustness of SP and the other best performing baselines further, figure 5.4 shows the difference in terms of ERR-IA between the QL method and the best performing baseline runs. The SP run behaves differently compared to the other approaches for negatively affected queries. For the SP run, the average negative difference was -0.0417 which means that SP hurts



(a) SP

(b) LDA

(c) K-means

(d) Spectral clustering

Figure 5.4: Relative changes between diversification methods and the QL baseline on the ERR-IA metric for the 200 test queries.

Approach	Help avg.	Hurt avg.
SP (xQuAD)	0.0972	-0.0417
Spectral clustering (PM-2)	0.0949	-0.1529
LDA (xQuAD)	0.0904	-0.1428
K-Means (Square loss)	0.0866	-0.1378

Table 5.5: The average of ERR-IA change between best runs and the QL baseline for helped and hurt queries.

queries slightly compared with the other approaches. Table 5.5 presents the average help and hurt differences for the best performing runs. As can be noticed from the table and figure, all approaches helped relatively similar number of queries at a quite comparable rate but the SP method is the only approach that manages to lower the impact of hurt queries.

These findings provide answers to my first research question RQ1. Statistically significant improvements were obtained when using social positions with the xQuAD and IA-Select diversification functions under various diversification evaluation metrics. All the other subtopic identification approaches failed to provide any statistically significant improvement over the QL baseline. The SP method also demonstrated a robust performance compared with the other baselines, particularly for negatively affected queries. These results suggest that social positions provide effective subtopics representation for diversification approaches.

5.6.2 Performance factors

In section 5.6.1, I validated the use of social positions as a subtopics representation for diversification algorithms. In this section, I focus on the second research question of this chapter by investigating factors that influence the performance of diversification based on social positions.

Firstly, the considered baselines are essentially clustering approaches which require setting the number of clusters, or terms for the DSPApprox approach, in advance. This number is static and applies for all queries. It is, perhaps, unrealistic to assume that all queries have the same number of subtopics. This is a standard issue with clustering methods which could affect the performance of diversification approaches since they, by definition, seek to cover all topics in the diversified list. Documents from irrelevant clusters would be promoted to the list. This issue does not apply to the SP method. In SP, the number of social positions for each query varies based on the matching process that is discussed in section 5.4.1. SP only considers social positions with matched documents. As can be seen in figure 5.5, about 78% of queries have between 1 to 4 matched social positions.

In figure 5.5, almost 28% of queries were assigned to one social position. About 70% of those queries are faceted queries. A facet query has one dominant interpretation but multiple subtopics, e.g. the previous example of *Harry Potter*. The assignment of one social position for such queries means that the diversification strategy, in section 5.4.2, aims to cover the multiple subtopics of such a single interpretation which is a desired behaviour. However, not all faceted queries were assigned one social position. In total, there were 142 faceted queries and some of them were assigned multiple but similar social positions; for example, *coffee addict* and *coffee lover* for the query *starbucks*. Another observation from figure 5.5 is that about 11% of queries were not matched to any social position. In this case, the top 50 documents were diversified as if they belonged to one single social position. In terms of diversification performance for these queries, 7 out of 23 queries were positively affected and the remaining queries were either hurt or unchanged

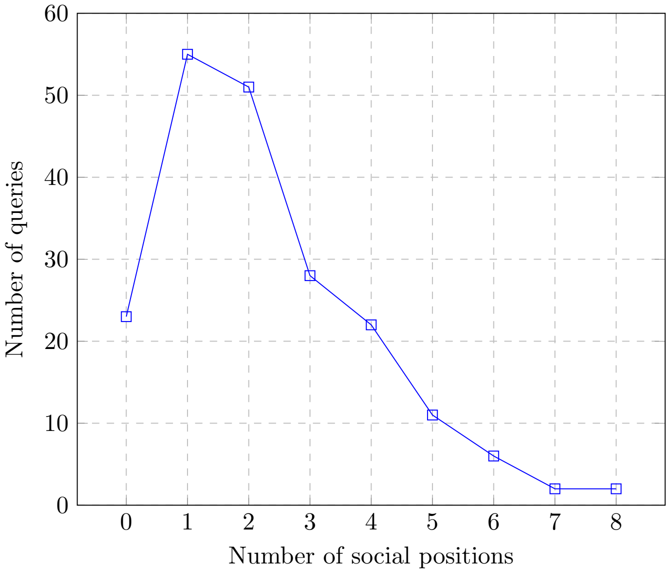


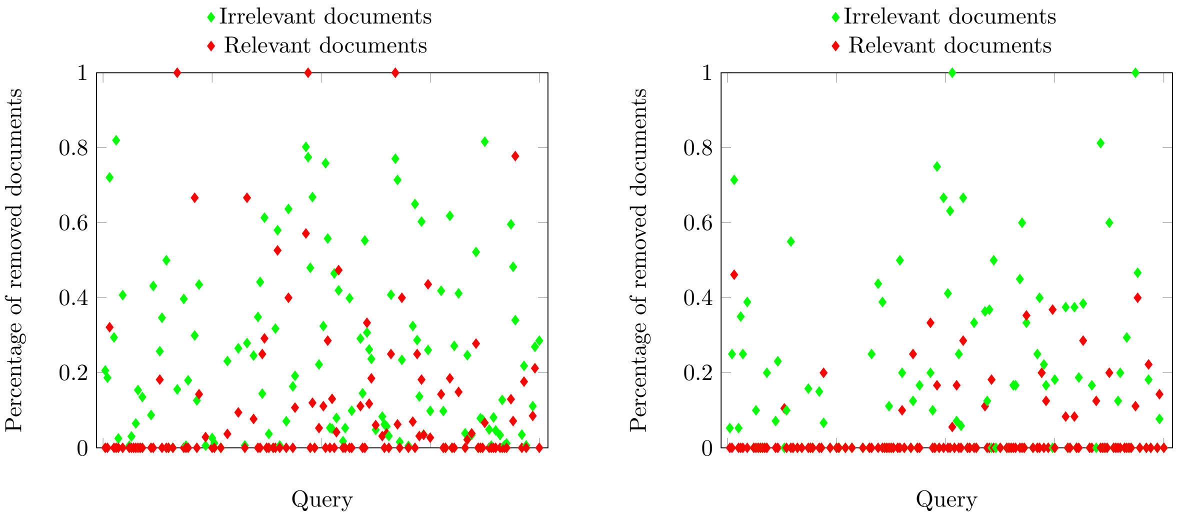
Figure 5.5: The distribution of assigned social positions per query.

	ERR-IA	α-NDCG	NRBP	S-Recall	+	-	=
QL	0.2798	0.3741	0.2433	0.5769			
Phrase	0.2885	0.3824	0.2522	**0.5778**	63	19	118
xQuAD	0.2908	0.3826	0.2561	0.5753	104	65	31
Square loss	0.2776	0.3728	0.2397	0.5690	103	73	24
PM-2	0.2838	0.3679	0.2514	0.5633	89	88	23
IA-Select	0.2999	0.3858	0.2713	0.5712	100	79	21
SP	$\mathbf{0.3268}^{\mathrm{P,X}}_{\mathrm{A}}$	$\mathbf{0.4063}^{\mathrm{A}}$	$\mathbf{0.3016}^{\mathrm{P,X}}_{\mathrm{A}}$	0.5642	107	68	25

Table 5.6: Search accuracy for various diversification algorithms when removing documents from which snippets cannot be extracted. I use letters to indicate statistically significant differences. P, X, and A refer to the Phrase, xQuAD, and IA-Select runs, respectively.

compared with the QL baseline. The average change is similar to that in table 5.5.

The second factor relates to the estimation of a social position's relevance to a query. In section 5.4.1, I defined a method for extracting snippets for a query based on the query's key phrase. Each snippet must contain the query's key phrase and at least one query's term. Documents from which snippets can be extracted are used to estimate social positions' relevance. In contrast, topic importance, $p(t|q)$, for the other baselines was estimated using all documents and not just those from which snippets can be extracted. Documents with snippets are more likely to be relevant to the query than the others. For a fairer comparison, table 5.6 presents the search accuracy for various diversification algorithms over documents from which snippets can be extracted. Subtopics were identified using the LDA approach over these documents. This filtering process allows all methods



(a) 200 documents.

(b) 20 top documents.

Figure 5.6: Percentage of removed relevant and irrelevant documents for each test query as a result of applying the similarly scoring function in equation 5.23.

to estimate topic importance using the same subset of documents that is used by the SP method. The *phrase* baseline has the same ranking as the QL method but documents without snippets are removed. The results suggest that such a process helps in improving almost all methods, including the SP approach. The rate varies from 1.82% for the xQuAD to 5.78% for the IA-Select compared with their performance in table 5.4.

Thirdly, it is important to examine the possible effect of the distance measure $S(d, r)$, as in equation 5.23, which I used to measure the relevance of a document to a social position. The SP method requires documents to have a distance score above or equal to 0.10 to be included in the diversified list of a search query's position. This threshold might be helping to remove irrelevant documents whilst keeping relevant ones for the set of documents representing a social position. Figure 5.6 plots the percentage of removed documents for each query. When considering the top 20 documents from the QL list, an average of 6 irrelevant documents were removed from the list for 62 queries compared with 4 relevant documents for 25 queries. Of the 62 queries, SP improved the performance for 36 and hurt 13 compared with the QL baseline. When considering all 200 documents in the initial list, the proportion of affected queries almost doubled for both cases. This suggests that the distance measure helped to improve the SP performance by removing irrelevant documents.

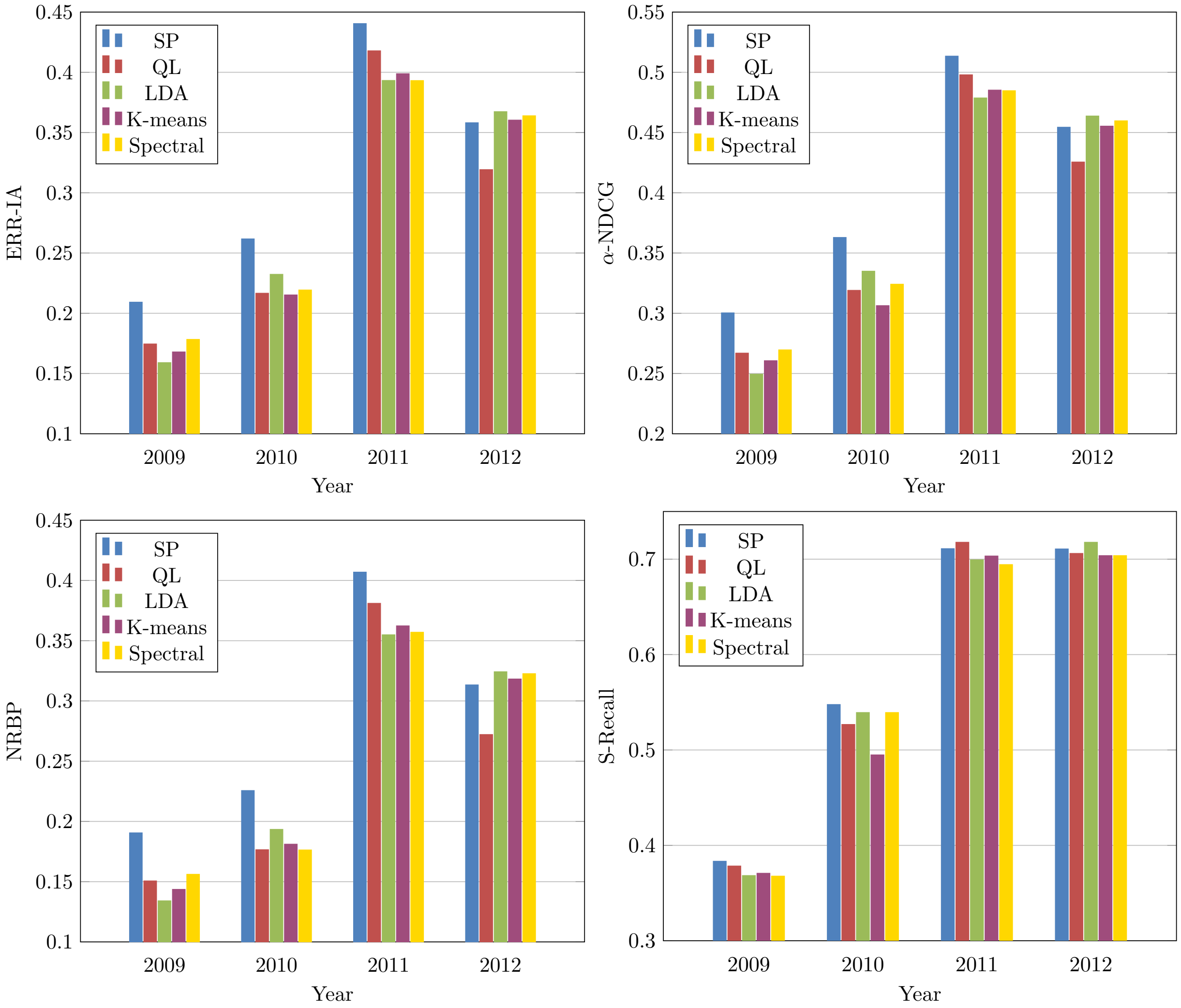


Figure 5.7: Search accuracy for each test collection. For TREC 2009 and 2010, queries were selected to be more frequent compared with those in TREC 2011 and 2012.

5.6.3 Improvement and failure analysis

Queries for the TREC 2009 and 2010 test collections were selected to be more frequent, medium to high frequency, compared to those in the TREC 2011 and 2012. In figure 5.7, I plotted the performance of the SP approach as well as the other baselines under ERR-IA, α-NDCG, NRBP, and S-Recall. All approaches seemed to perform well for obscure queries compared with popular ones. The SP approach has the highest relative improvement over QL on TREC 2009 and 2010 by 19.89% and 20.82% under ERR-IA, respectively. In terms of α-NDCG, the relative improvement compared with QL is 12.51% and 13.76% for TREC 2009 and 2010. The relative improvement for SP over QL dropped for obscure queries to 5.41% under the ERR-IA for TREC 2011. In addition, the SP performance was close to the other baselines for less frequent queries. Similar trends are observed for both NRBP and S-Recall.

Another dimension investigated was the performance of all considered approaches on

	ERR-IA	α-NDCG	NRBP	S-Recall	+	-	=
Faceted							
QL	0.3135	0.4097	0.2753	0.6205			
LDA	0.3207	0.4156	0.2834	0.6238	79	44	18
K-means	0.3194	0.4108	0.2846	0.6043	79	48	14
Spectral clustering	0.3283	0.4237	0.2928	0.6157	82	41	18
SP	$\mathbf{0.3458}^{Q}$	$\mathbf{0.4360}^{Q}$	$\mathbf{0.3107}^{Q}$	**0.6308**	70	45	26
Ambiguous							
QL	0.1974	0.2872	0.1648	**0.4701**			
LDA	0.1997	0.2869	0.1672	0.4580	26	15	16
K-means	0.1953	0.2832	0.1636	0.4624	30	16	11
Spectral clustering	0.1840	0.2776	0.1495	0.4609	26	18	13
SP	$\mathbf{0.2385}^{Q,S}$	$\mathbf{0.3261}^{Q}$	$\mathbf{0.2113}^{Q,S}$	0.4649	27	16	14

Table 5.7: Search accuracy for faceted and ambiguous queries. Statistical significance to QL, LDA, K-means and Spectral clustering are denoted with Q, L, K and S, respectively.

faceted and ambiguous queries. The results are shown in table 5.7. SP was the only approach that provided statistically significant improvements over the QL baseline for both types of query. The table shows that SP performed better with ambiguous queries than on faceted queries. For faceted queries, SP improved by 10.30%, 6.42%, and 12.86% under evaluation metrics ERR-IA, α-NDCG, and NRBP, respectively. These relative improvements were doubled for ambiguous queries. SP works by identifying the multiple possible interpretations for a query via the identification of the query's social positions then documents are diversified for each social position based on its LDA's subtopics. The successful performance with both types could suggest that the identification of a query's social positions works effectively. This is particularly true for ambiguous queries, which seem to be challenging for all the other baselines. Faceted and less popular queries seem to be easier to diversify using the other baseline approaches as shown in table 5.7 and figure 5.7 given that TREC 2011 and 2012 are predominantly composed of faceted queries (about 80%). These results provide answers to my third research question RQ3 regarding the performance of SP on different query types.

5.7 Summary

In this chapter, I presented a search results diversification approach based on social positions. Diversification is a strategy that deals with ambiguity in search queries which is a common issue as reviewed in section 5.1. Diversification approaches, as discussed in section 5.2, require a set of subtopics for each query in order to produce a diverse list of results. Clustering approaches are typically used to construct a set of subtopics for each query in cases where explicit subtopics are not available. In section 5.4, I introduced an approach to use social positions as a representation of the multiple possible subtopics of a search query. Firstly, as in section 5.4.1, the user models that were estimated in the previous chapter were used to match a search query to its most relevant social positions. Secondly, as in section 5.4.2, a diverse list of documents was built for each candidate social position using a diversification function. These lists of documents form an input to a selection strategy that proportionally diversify the final ranked list based on the importance of the query's social positions. The experiments described in section 5.6 with multiple test collections demonstrated the effectiveness of using social positions as a topic representation for diversification compared with various widely used clustering techniques.

CHAPTER 6

Framework validation II: Session-based search

In this chapter, I validate my proposed user modelling framework using the task of session search. In the context of my experiments, a search session is a sequence of interactions by a user to fulfil a single information need (Jansen et al., 2007a). Examples of such interactions include query reformulations and clicks on search results. The goal of the session search task is to design retrieval systems that take previous user's interactions into account when presenting results for the user's next query within the same session (Kanoulas et al., 2010). I use this task to test the validity of the user modelling framework I presented in chapter 4. For this purpose, I consider session search as an example of short-term personalisation. The incorporation of the previous user's interactions means that the results for the user's next query will be personalised to his/her current interest. As discussed previously, personalisation often requires some form of a user model. The goals of my proposed framework in the context of session search are twofold: (1) to detect and represent the user's social position within a single search session; (2) to use the identified social position as a user model in personalising the search results for the user's next query. In section 6.1, I discuss motivation for session-based search followed by a review of related areas in section 6.2. Approaches to evaluating session search systems are presented in section 6.3. The proposed approach is detailed in section 6.4. Sections 6.5 and 6.6 present the experimental settings and results, respectively.

6.1 Motivation

Web search can take two forms of interaction between the user and the search engine: static or dynamic. In static settings, the search engine presents a list of items in response solely to the submitted query. If the user submits a follow-up query, the process is repeated. Any feedback from the user in such settings is not used. In realistic situations, search is more likely to be dynamic where users interact with the results provided and reformulate their original query. This type of interaction can be considered as a form of feedback. Based on early studies of search logs, approximately 37% of users submit at least one modified request of their initial query. A figure of 36.3% from the AltaVista dataset is reported by Silverstein et al. (1999) while Aloteibi and Sanderson (2014) found a similar statistic (37.9%) using the 2006 Microsoft Live Search logs. Other studies have reported various figures ranging from 28% using the AOL logs (Pass et al., 2006) to 45% on the 2001 Excite dataset (Wolfram et al., 2001). Considering the sheer size of search queries submitted daily to search engines, query reformulation is responsible for a sizeable portion of search traffic and is one of the widely studied areas of users' behaviour in web search (Anick, 2003; Huang and Efthimiadis, 2009; Jansen et al., 2009; Lau and Horvitz, 1999).

Reformulation might occur for contrasting reasons. It could indicate either a struggle in satisfying an information need or a success in locating relevant information for a specific aspect of a multi-facet information need and a move into researching another aspect (Hassan et al., 2014). Either case is usually accompanied by behavioural actions that along with the reformulation sequence can signal latent variables about the user. For example, one important user action occurs when users click on a result and spend some time, known as *click dwell time*, examining the clicked item. Joachims (2002) has suggested using clickthrough data as a substitute for explicit relevance judgments when building learning to rank models for which large-scale training data is difficult to construct. A later study by Fox et al. (2005) has found an association between click dwell time and user satisfaction. The longer the user stays on a clicked item, the more likely that it is to satisfy their need. Researchers have also used other behavioural actions such as mouse movement (Guo and Agichtein, 2008; Huang et al., 2012).

Although these behavioural signals are too noisy in nature to be used as implicit relevance feedback, they present an opportunity to integrate unobtrusive users' behavioural information into various search engines' components. For instance, Agichtein et al. (2006) incorporated clickthrough and browsing features into ranking models and showed that it could provide significant improvement. The work in this chapter follows a similar assumption. Users' actions, namely reformation sequence and clickthrough data, might help in personalising search results during a session to the user's current interest. In the context of my user modelling framework, such behavioural signals are used to identify which social position the user is taking during the session. The previously built models for each social position would then provide additional features to be used in personalising the search session. The overall goal is to make the search experience stateful rather than its current stateless situation.

6.2 Background

The behaviour of information seekers has been the focus of several studies using a variety of methods such as: query log analysis, controlled user experiments, and simulation.

These studies attempt to explain or model some aspects of users' behaviour during search. In order to differentiate between *good* and *less successful* search strategies, the users population is often divided into two groups based on criteria such as expertise (Aula, 2003; Hölscher and Strube, 2000), search outcome (Aula et al., 2010; Hassan et al., 2014; Odijk et al., 2015), completion speed (Aula and Nordhausen, 2006) or the difficulty of the search task (Singer et al., 2012). Understanding how and why users, in either group, make certain decisions during the search process promises to aid development of search engine components that are user-centric and dynamic. Among such components is the personalisation of a search session, which is the topic of this chapter. This topic is closely related to two aspects of Interactive IR[1]: query (re)formulation and examination behaviour which are reviewed in section 6.2.1 and section 6.2.2, respectively. I also provide an extensive review of session search systems in section 6.2.3.

6.2.1 Query (re)formulation

Navarro-Prieto et al. (1999) identified three strategies of web search: top-down, bottom-up, and mixed. In a top-down approach, users tend to start with a broad query and then specify it based on their interaction with the results until relevant information is found. Users following the bottom-up strategy will start with specific queries usually derived from a task description. The study found that experienced users tend to follow a bottom-up approach more often in fact-finding tasks than novices who are associated with the top-down approach. Experienced users were also shown to alternate between these two strategies, a mixed approach, and their search behaviour is more planned and structured compared with novices. Another user study was conducted by Aula (2003) to investigate initial query formulation strategies. The results indicated that users experienced in web search issued long and more precise queries compared with the novice ones. This study also noted that although the majority of search sessions consist of a single query, it does not necessarily mean users have succeeded in locating relevant information as some might have exhibited different stopping behaviour than others. Some users may patiently decide to go through the list of results to find relevant documents.

Several taxonomies of reformulation have been introduced in the literature based on lexical, syntactic, or semantic interpretation of reformulation patterns in query logs (Anick, 2003; Boldi et al., 2009; Bruza and Dennis, 1997; Guo et al., 2008; Huang and Efthimiadis, 2009; Jansen et al., 2007b; Lau and Horvitz, 1999; Rieh and Xie, 2006; Teevan et al., 2007a). An analysis conducted by Anick (2003) suggested that nearly two thirds of refinements constitute one of the following types: modifier, head, and elaboration. All of these types

[1] In fact, session search is also related to other areas of IIR such as stopping behaviour (e.g. Maxwell et al., 2015) and interaction cost (e.g. Azzopardi, 2011). However, the experiments I conducted are based on TREC Session search tracks, which considers query reformulation and results interaction data.

are results of the user adding terms to the original query that functions as a modifier, as a head or simply adds further context to the query. The sample size was, however, quite small (100 refinements). Huang and Efthimiadis (2009) took a lexical approach in studying reformulation by developed a rule-based classifier to map reformulated queries into one of 11 pre-defined categories. Reformulation effectiveness was evaluated based on users' click behaviour. A successful strategy is the one that results in a click. They found that spelling correction, expanding acronyms, and adding word reformulations are more likely to lead to a click if the user did not click on any results for the initial query. If the initial query resulted in a click, users' successful reformulation strategies are: word substitutions, word reordering, and adding words.

Boldi et al. (2009) performed a semantic classification study over large datasets of query logs. They considered four categories: generalisation, specialisation, error correction, and parallel move[2]. This study used a UK dataset representing users of the Yahoo UK search engine, and a US dataset. Users were more likely to perform parallel move in both datasets, at 48% and 56%, respectively. The next refinement categories were: specialisation (38 - 30%), error correction (10-5%), and generalisation (4-10%). In sessions where users submitted five or more queries, a slight increase in the parallel move and generalisation categories was observed at the expense of specialisation and error correction in the US dataset.

Sloan et al. (2015) applied a term-based methodology to study query reformulation in session search using data from the TREC Session tracks. They considered three term actions: retention, addition, and removal. One of their main findings was that users' choice of query terms changed progressively during the session. While, on average, two thirds of terms in a search query were retained in its successive reformulation, users ended their session with different terms compared with the initial query. They also found that in long sessions (5 queries or more) users were more likely to shift their search to a parallel aspect, which is similar to the findings of Boldi et al. (2009). These long sessions are perhaps exploratory in nature (Marchionini, 2006). Jiang and Ni (2016) followed a similar term-based methodology by using data collected in controlled user study settings. They studied three types of users' strategies: to remove or retain a word in the current query, to add a new word, or to re-use a previously removed word. Their results suggested that users were more likely to remove words for two reasons. First, the removed term represents an aspect of the query and the user has already located relevant information (by means of a satisfying click). Second, the removed term might not have improved the relevance of the results. Users were more likely to remove off-topic terms that did not frequently co-occur with the query's other terms or occurred less in the results. They also showed that users were likely to source new terms for their next query from the results' titles and

[2] A modification of the original query from one aspect to another.

snippets of the current query regardless of whether they clicked on these results or not.

Another source of reformulated queries can be the query suggestion component. Most search engines present a list of suggested queries to the user as a static list in the user interface or as an auto-completion service. The generation of suggested queries has been the focus of a number of studies using query logs, browsing behaviour, and other data sources (e.g. Baeza-Yates et al., 2004; Beeferman and Berger, 2000; Cucerzan and White, 2007; Dang and Croft, 2010; Dehghani et al., 2017; Sordoni et al., 2015). Of particular interest is the study of Kato et al. (2013) who investigated the use cases of query suggestion. They found that users were more likely to use this feature when the initial query is rare or a single-term query. They also found that users were more likely to click on a suggested query to generalise their initial query. The rarity of the search query has been found to negatively affect the retrieval effectiveness and lead to users reformulating their query. Downey et al. (2008) also found that users might need to submit more reformulated queries to satisfy a rare information need than they do with common goals. Users would also adapt to the degraded performance of the search engine by submitting more queries (Smith and Kantor, 2008).

Previous research has also focused on situations where users struggle to find relevant information. Aula et al. (2010) found that users' reformulation strategies tend to be unsystematic when they experience difficulties in finding relevant information; a behaviour most often exhibited by novices than experts who reformulate systematically (Aula and Nordhausen, 2006). They also reported that struggling users issue more queries per session and spend longer time examining the results page, which has also been found by Singer et al. (2012). Hassan et al. (2014) argue that the length of a session is not necessarily a sign of a struggling user as the user might be researching a multi-facet information need, i.e. an exploratory session. They showed that struggling users tend to compose reformulated queries that are semantically or lexically similar to the initial query while exploring users will deviate slightly from the initial query. Signals based on click behaviour and dwell time were also shown to correlate with struggle under some constraints. Their labelling of 3000 sessions indicates that 40% of users are exploring, 23% are exploring with struggle, and 36% are struggling.

An important dimension of the search process relates to the user's cost of formulating, reformulating their queries, or examining search results. Azzopardi (2009) provides some justifications for the shortness of search queries, which is evident in query logs (e.g. Wolfram et al., 2001). While this study showed that long queries perform effectively in terms of retrieval performance, users decision to submit short queries in the range of 2 to 5 terms may be supported by the Law of Diminishing Returns. Gain in terms of retrieval effectiveness diminishes as more terms are added to a short query. Keskustalo et al. (2009) and Baskaya et al. (2013) concluded that sessions of short queries would often lead to a satisfactory outcome. In a laboratory study, Azzopardi et al. (2013) found that as the physical cost[3] of querying increased, subjects would submit significantly fewer queries and examine more results.

6.2.2 Examination behaviour

After submitting a query, the user is expected to interact with the Search Engine Results Page (SERP). This interaction has been vastly studied in terms of clicks as the most salient and easily captured user behaviour. Typically, a set of hypotheses, known as *click models*, about how users examine the SERP are generated and validated (Chuklin et al., 2015). These click models are intended to simulate real users' interactions with the SERP by modelling their observed behaviour in click logs. In reality, the observed users' behaviour is not ideal and is often biased. An eye-tracking study by Joachims et al. (2005) showed that results at rank 1 and 2 receive the most attention from users and that users' viewing time falls as the document rank decreases. In terms of clicks, relevance seemed to not be the only factor that affected users' click decision. Users were more likely to click on the first document than the second even though they both receive similar attention and the second document is more relevant than the first. Joachims et al. attributed this behaviour to a bias in a user's decision to *trust* the search engine ranking. This behavioural bias, which is also known as *position bias* or *presentation bias* (White, 2016), means that the probability of clicking on a document depends on its perceived relevance and its position in the SERP. Documents at higher ranks are much more likely to be clicked on than those at lower ranks. However, such an assumption is over simplistic because the user's click may depend on the relevance of higher ranked documents and not just the document position (Chapelle et al., 2009).

A family of click models has been developed to better explain users' behaviour. Craswell et al. (2008) introduced the cascade model in which they assume that users examine the SERP from top to bottom and stop once they find a relevant document. In such a model, users are assumed to examine each document snippet to decide whether or not to click. A click on document at rank r indicates that all documents up to rank r were examined and are not relevant. Extensions to the cascade model have been built to account for various scenarios such as sessions with multiple clicks (Guo et al., 2009), users not being satisfied after a click (Chapelle and Zhang, 2009) and users skipping snippets without examination (Dupret and Piwowarski, 2008).

Several other biases have been shown to affect users' behaviour and consequently their click patterns (White, 2016). *The domain bias* is where documents from certain domains have been shown to attract users (Ieong et al., 2012). *The caption bias* is where certain properties of document's title and snippet may capture users' focus (Clarke et al., 2007;

[3] Physical cost was measured by time needed to submit a query.

Yue et al., 2010). *The cognitive bias* is where users may be influenced by cognitive variables such as preferring positive content over negative one in the context of health-related search (White, 2013). *The attention bias* applies for modern SERPs where it is becoming standard to include matched items from other verticals such as news, images, and videos within the SERP (Chen et al., 2012a; Wang et al., 2013). This bias accounts for observations where users are more likely to examine the vertical and documents positioned around it. Chuklin et al. (2015) provided a more detailed discussion of click models under a variety of examination patterns.

6.2.3 Session-based search systems

The goal of session search is to improve retrieval performance over a single search session. Each search session consists of multiple queries that are submitted by a user to fulfil a single information need. These needs can be exploratory in nature or based on fact-finding tasks (Marchionini, 2006). The assumption is that users' interactions by means of reformulation and examination behaviour can be utilised to achieve such a goal. There are a number of ways in which session search can be approached. One is to model the interaction process as a sequential decision making process where the goal is to learn a policy that maximises a pre-defined reward eventually leading to an improvement in the retrieval performance. Previous work in this direction models session search using the framework of Markov Decision Processes (MDP) (Chen et al., 2018; Guan et al., 2013) or its variant known as Partially Observable MDP (POMDP) (Luo et al., 2015a, 2014b; Yang et al., 2018). According to Puterman (2014), MDP can be described as a tuple of $\{T, S, A_s, P_t(.|s, a), r_t(s, a)\}$[4] where:

- T is a set of decision epochs. In session search, decision epochs are finite (i.e. number of reformulated queries is finite and observed).
- S is a set of system states at each decision epoch.
- A_s is a set of actions that is available to the agent at state s.
- $r_t(s, a)$ represents the agent reward, or cost, after taking action a at state s.
- $P_t(.|s, a)$ represents the probability distribution that determines the state to which the system will transition after the agent takes action a at state s.

The aim is to learn an optimal policy that maximises the agent's reward. A policy prescribes action selection for the agent at each state (Puterman, 2014).

Guan et al. (2013) proposed the Query Change Model (QCM) as a session search retrieval model. QCM is based on MDP and treats queries as the system states. In this

[4] In some notations, a discount factor γ can be introduced to discount future rewards.

model, there are two agents: the user and the search engine. The user's actions are term retention, removal or addition while the search engine's actions are based on increasing, decreasing or maintaining query terms' weights. The authors designed specific policies for the search engine to handle a set of pre-defined scenarios. Examples of such scenarios are: increasing the weight of added terms that the user did not source from the snippets of previous query's clicked documents and decreasing the weight of removed terms that appeared in the previous query's SERP or clicked documents. I use QCM as a baseline in this chapter. Luo et al. (2014b) assumed that the system states are hidden and thus model session search as a POMDP. Their proposed model, called Win-Win search, uses four hidden states based on two dimensions: relevance and exploration. One possible state is when the user might have found the returned results for query q_{i-1} relevant but decided to continue researching the same information need in the subsequent query q_i by exploiting some terms from the SERP and clicked documents of q_{i-1}. Since such states are hidden, the system maintains a probability distribution over the set of possible states based on observations that the search engine makes from the user's clicking behaviour or term-based changes in reformulation. The Win-Win system selects actions for the search engine from 20 options that are based on various configurations of term weighting or retrieval models.

Luo et al. (2015a) also modelled session search using POMDP but learnt optimal policies directly from a set of features describing the observations that the search engine can make from the interaction process. Luo et al. (2015b) explores various choices for selecting states, actions, and rewards in the design of retrieval models based on MDP. They conclude that Win-Win configuration provides the best performance but is not time efficient compared to other instantiations such as the QCM. Chen et al. (2018) proposed a multi-agent MDP model where each agent is trained to rank documents for a specific cluster of related queries. The number of clusters and the number of MDP agents are determined according to a model based on the Chinese Restaurant Process framework. In such a model, states are defined as the set of documents available to the agent to choose from in order to fill a specific position in the ranked list.

One issue with all of these MDP-based solutions is the choice of the reward function. They rely on using the ground truth judgments provided by NIST assessors in their reward function. Typically, reward is defined in terms of nDCG@10 and the policy that maximises the reward is used to rank documents. In practice, however, relevance assessment labels are not available to the agent therefore, the performance of such systems might be seen as an upper bound. A more realistic reward function can be defined based on SAT clicks (satisfactory clicks with dwell time above 30 seconds) as demonstrated by Guan et al. (2013), Luo et al. (2015a), and Luo et al. (2015b). However, performance deteriorates substantially when using SAT clicks instead of the ground truth judgments in the definition of the reward function.

The second class of session search systems focuses on query formulation. Since session-based search systems are evaluated based on their ability to improve the retrieval performance for the last query in each test session (Carterette et al., 2016), they are expected to use the query chain, previous ranked documents, and click information to accomplish this task. Also, document relevance is judged using the entire query chain of the session and not just the last query. Therefore, one feasible approach is to use such information to compose a new query that better represents the user's information need compared with the last query. Guan et al. (2012) extracted *nuggets* from the query chain and used them to formulate a new weighted query. *A nugget* for query q is a sequence of query terms that appear within a short distance of each other in the snippets of query q. Albakour et al. (2010) expanded the last query with terms extracted from anchor text. Expansion terms might originate from anchors linking to results that were presented to the user during the interaction process or from links pointing to clicked documents. Guan et al. (2012) also used anchor text in a similar way.

Guan and Yang (2014) explored various methods to aggregate all queries in a test session into a single query. Their best performing scheme calculates the relevance of document d to test session s as in the following equation:

$$score(s, d) = \sum_{i=1}^{n} \gamma^{n-i} score(q_i, d) \tag{6.1}$$

where q_i is the i^{th} query of session s. The discount parameter γ is estimated using a parameter sweeping method and found to be optimal at ($\gamma = 0.92$) which means the last query in the session receives the highest weight and then weight decreases progressively towards the first query. Such a discount has been previously used to decrease the weight of previous actions (e.g. Bennett et al., 2012; White et al., 2010). Van Gysel et al. (2016) concatenated all queries from the query chain into one query.

A widely used statistical measure of relevance is based on the negative KL-divergence between a query language model θ_q and a document language model θ_d as in the following equation (Lafferty and Zhai, 2001; Nallapati, 2006):

$$\begin{aligned} score(q, d) &= -D_{KL}(\theta_q \parallel \theta_d) \\ &\overset{rank}{=} \sum_{w} P(w|\theta_q)\ log\ P(w|\theta_d) \end{aligned} \tag{6.2}$$

In the QL model, the query model θ_q is estimated using Maximum Likelihood Estimation (MLE) as in the following:

$$P_{MLE}(w|\theta_q) = \frac{c(w, q)}{|q|} \tag{6.3}$$

where $c(w, q)$ is the count of occurrences for term w in query q. In session search, the query chain and click data present additional contextual information that can be incorporated to estimate a new query model $\widehat{\theta_q}$. One popular approach to estimate $\widehat{\theta_q}$ is the Fixed Coefficient Interpolation (FixInt) method (Shen et al., 2005). As the name implies, FixInt interpolates between two language models: the current query model θ_q and a history model H. The history model H is also an interpolation between a click history H_c and a query history H_Q. Let $H_Q = (Q_1, \ldots, Q_{n-1})$ represent the session queries prior to the current query Q_n and $H_C = (C_1, \ldots, C_{n-1})$ is the click history of the session where C_i is the concatenation of the titles and snippets of clicked documents at step i. The new query model $\widehat{\theta_q}$ is estimated as:

$$P(w|\widehat{\theta_q}) = \alpha P_{MLE}(w|\theta_q) + (1-\alpha)P(w|H) \tag{6.4}$$

where,

$$P(w|H) = \beta P(w|H_C) + (1-\beta)P(w|H_Q) \tag{6.5}$$

$$P(w|H_C) = \frac{1}{n-1}\sum_{i=1}^{n-1} P_{MLE}(w|C_i) \tag{6.6}$$

$$P(w|H_Q) = \frac{1}{n-1}\sum_{i=1}^{n-1} P_{MLE}(w|Q_i) \tag{6.7}$$

The FixInt method and its variants have been shown to perform effectively in the TREC Session tracks (Jiang and He, 2013; Jiang et al., 2012; Levine et al., 2017). For example, Jiang et al. (2012) built the new query model $\widehat{\theta_q}$ by interpolating the current query model θ_q, the query history H_Q, and a relevance model R as in the following equation:

$$P(w|\widehat{\theta_q}) = (1-\lambda)\big((1-\beta)P_{MLE}(w|\theta_q) + \beta P_{MLE}(w|H_Q)\big) + \lambda P(w|R) \tag{6.8}$$

The relevance model R is estimated using relevance modelling (Lavrenko and Croft, 2001) for a query model constructed using equation 6.8 but with $\lambda = 0$. Levine et al. (2017) constructed a query model $\widehat{\theta_q}$ by inductively interpolating a model of the current search iteration with its preceding one up until the test query q_n. Both Shen et al. (2005) and Levine et al. (2017) suggested methods to dynamically set interpolation parameters. Li et al. (2018) estimated the query model using Markov Random Field while Zhang et al. (2016a) applied an expansion model for session search based on the idea of photon polarisation from the field of Quantum Theory. White et al. (2010) constructed three models to predict users' future interests within a single search session. These are: a query model to represent the test query, a context model for the user's previous interactions during the session[5], and an intent model as the linear combination of the query and context

[5] This includes previous queries, clicked documents and subsequent navigation after clicking on a

models. The linear combination parameter was learned for each query based on various features. Each model is a multinomial distribution over the Open Directory Project (ODP) topic categories. They showed that the intent model could predict a user's future interests in the same session with reasonable accuracy. Others have addressed session search as a diversification task where the goal is to produce a rank list that covers multiple aspects of the user's information need (Raman et al., 2013).

The third category of systems, including the approach I introduce in this chapter, views session search as a learning to rank task. The utilisation of interaction data to formulate a new query or build an expansion model represents a single source of belief, or feature, about a document's relevance. However, it has become standard for web search engines to rank documents based on multiple features rather than fully relying on one scoring function or assuming a linear combination of features. Session search is no exception and many features can be extracted from the interaction information. Bennett et al. (2012) considered three temporal views of a user's interaction. The first is a session view to capture a user's interactions within the current session. The second is a historic view that covers interactions prior to the current session and the third is an aggregate view. They defined a unified set of features calculated based on the three views and used LambdaMART as their LTR algorithm. Their results suggest that the historic view provides a significant improvement in personalising the initial query of search sessions. As the session progresses, gain provided by the session view increases while the benefits of historic information decrease. Liu et al. (2012) conducted a laboratory study designed to identify a document's usefulness, or relevance, predictors. The most indicative predictor was found to be dwell time of clicked documents. They built decision tree models using dwell time and a few other variables to predict relevance. Documents that were judged as relevant were then used to extract expansion terms.

Zhang et al. (2016b) noted that previous work mostly assumed that recent queries, or contextual models built using recent queries, are more important than older interactions (e.g. the exponential decay parameter in equation 6.1). They proposed several hypotheses to weight contextual models. For example, if an interaction results in a SAT click then its contextual model's weight should be increased. They cast the aggregation of the multiple contextual models extracted from consecutive interactions as a LTR problem. Several other studies have applied LTR algorithms to the task of personalising results during a search session using various features (Shokouhi et al., 2013; Ustinovskiy and Serdyukov, 2013; Xiang et al., 2010) and within the TREC Session tracks (Chen et al., 2012b; Jiang and Allan, 2014; Xue et al., 2014).

In this chapter, I present a session search system belonging to the LTR category. Similar to previous work, I use features based on the current query and the search session.

document.

The main contributions of this chapter are:

- I introduce a method to map a test session to its most relevant social positions. A user model represents each social position. The user model consists of terms that are relevant to both the session and the identified social position.
- I define a set of novel LTR features derived from a search session's social positions.
- I identify related sessions from query logs using social positions' features. These related sessions are used to source novel LTR features.

6.3 Evaluation

In Cranfield-like experimental settings, the input to IR systems is only a list of test queries. These systems are evaluated using a variety of test metrics computed over their resultant ranked lists of documents. In session search, and more generally personalisation research, the difference is primarily about the input being not just a list of typically short queries but also some additional contextual information about the current user. The output is still a ranked list of documents for each query that can be evaluated using standard test metrics given some relevance judgments. In section 6.3.1, I discuss the TREC Session tracks (Carterette et al., 2016) that provides the primary test collection available for session search experiments. These test collections use the standard evaluation metrics discussed in chapter 2.

A number of notable session based evaluation metrics have been introduced in the literature. Järvelin et al. (2008) proposed the Session-based DCG (sDCG) for multi-query sessions. sDCG first starts by computing a DCG score for each query in the session and discounting it by $(1 + log_{base}(i))^{-1})$ where i is the position of the query in the session. A higher base of the log function suggests a small discount and a patient user who is willing to reformulate whilst a small base indicates the opposite. sDCG can be normalised in a similar way as DCG and ERR. Yang and Lad (2009) noted that sDCG makes a deterministic assumption about users' behaviour that is not realistic. For example, DCG for each query is computed over a fixed rank, $k = 10$, implying that the user will read all of the k documents for each query in the session. They suggest calculating the Expected Global Utility over a set of possible interaction patterns. Kanoulas et al. (2011a) introduced a suite of model-based and model-free session evaluation measures. Tang and Yang (2017) investigated the challenge of calculating an upper bound for a number of session search metrics and suggested a new normalised measure.

6.3.1 Test collections

As discussed in section 2.7, there are three components for each test collection: a document collection, topics, and relevance judgments (Sanderson, 2010). In session search, and more generally personalisation, a fourth component about the user's context is included. In this section, I describe each of the four components in the context of test collections developed and used by the TREC Session tracks from 2011 until 2014 (Carterette et al., 2013, 2014; Kanoulas et al., 2012, 2011b).

The TREC Session tracks of 2011 and 2012 used ClueWeb09, described in section 5.3.2, as their document collection and in the tracks of 2013 and 2014, ClueWeb12 was used. ClueWeb12 is the successor of ClueWeb09. It contains about 733 million English web pages crawled in early 2012 (Callan, 2012). A subset of about 52 million pages is referred to as ClueWeb12B. In terms of topics, each one is composed of a title, description, and narrative. Topics were created using TREC 2009 Million Query track queries (Carterette et al., 2009) and TREC 2007 Question Answering track questions (Dang et al., 2007). Organisers selected faceted topics from these two resources that are likely to require multiple reformulations to satisfy the full information need behind them. In the test collections from 2012 until 2014, a classification based on the search task was provided (Li and Belkin, 2008). Specifically, each topic was labelled based on two facets of search tasks: *product* and *goal quality*. A product search task can be intellectual when it results in new ideas or findings or factual when the result is locating specific information. The goal quality facet refers to whether the information need is well-defined or vague.

The set of developed topics were used to collect search sessions. A session contains the contextual information summarising the user's interaction with the search engine to fulfil the information need as specified in the topic. As stated earlier, session search aims to improve the retrieval effectiveness for a test query using some contextual information about the user. This context component of the TREC Session tracks comprises three types of information. The first is the sequence of queries leading up to the test query for each session. The second is the ranked list of documents for each past query, where a document is represented by its URL, title, and snippet. The third is users' clicking behaviour. Click information includes the order of a click and time spent on the document, i.e. dwell time. Sessions were collected in a crowdsourcing manner from actual users who were presented with a sample of topics to choose from and performed a search using custom-built tools. Note that each topic can be associated with more than one session. The relevance of a document was judged based on the topic description, i.e. the whole session relevance. The 2011 collection provides additional relevance judgments based on subtopics. Figure 6.1 shows an example session from the Session track 2012 and table 6.1 provides some statistics about these test collections.

The Session track was initiated in 2010 with a different context component (Kanoulas

```
<session num="1" starttime="09:54:08.725624">
   <topic num="1" product="factual" goal="specific" tasktype="known-item">
      <subject num="1">403b</subject>
      <desc>You are writing a summary article about US tax code 403(b) retirement plans.
       Find as many relevant documents as you can that would help you in writing the summary.
       Aspects might include eligibility for a 403(b), tax benefits of 403(b) plans,
       the types of institutions that offer them to employees, withdrawal rules, contribution limits,
       instructions for rolling over into another retirement plan, and so on.</desc>
   </topic>
   <interaction num="1" starttime="09:54:27.674484">
      <query>US tax code 403 (b)</query>
      <results>
         <result rank="1">
            <url>http://en.wikipedia.org/wiki/403(b)</url>
            <clueweb09id>clueweb09-enwp00-09-05733</clueweb09id>
            <title>403(b) - Wikipedia, the free encyclopedia</title>
            <snippet>The Employee Retirement Income Security Act (ERISA) does not require 403(b) plans to be
            technically qualified plans, i.e., plans governed by US Tax Code 401(a), but have ...</snippet>
         </result>
         ... 9 more results removed
      </results>
      <clicked>
         <click num="1" starttime="09:54:42.635247" endtime="09:54:43.722679">
            <rank>2</rank>
         </click>
         <click num="2" starttime="09:54:44.273290" endtime="09:54:45.208850">
            <rank>1</rank>
         </click>
      </clicked>
   </interaction>
   <currentquery starttime="09:55:00.877477">
      <query>US tax code 403 (b) eligibility</query>
   </currentquery>
</session>
```

Figure 6.1: An example from TREC 2012 Session track. Note that there are four task types depending on the product and goal of the topic. The known-item refers to a factual task with a specific goal. The known-subject is a factual task with an amorphous goal. An intellectual topic with a specific goal is interpretive while with an amorphous goal is called exploratory.

	2011	2012	2013	2014
# topics	62	48	49	60
# sessions	76	98	87	1021
# queries	280	297	442	4226
# sessions with clicks	64	72	82	641
# sessions with SAT clicks	46	46	51	444
Avg. session length	3.68	3.03	5.08	4.14
Avg. clicks per session	2.40	2.77	4.70	1.65
Avg. # sessions per topic	1.23	2.04	1.78	17.02

Table 6.1: Summary statistics of TREC Session tracks. Note that only the topics corresponding to the first 100 sessions of TREC 2014 test collection received relevance judgments.

et al., 2010). Each session contained only two queries without the rich interaction information available in subsequent test collections. A related TREC track has focused on dynamic domain search (Yang et al., 2017). A dynamic search system is initiated by a faceted query for which it should return a small set of documents. Next, a simulated user responds in real-time with detailed relevance feedback. The search system would then decide to exploit the relevance feedback to provide another set of documents to the user or stop the process. This track evaluated results on a passage level.

6.4 Learning to personalise for web search sessions

In this section, I detail the four main components of the framework. In section 6.2.3, I classified previous work in session search into three main categories based on their scoring methods: reinforcement learning, query formulation, and learning to rank approaches. The approach presented in this section falls into the learning to rank category. It is common for all of these approaches, especially for learning to rank methods, to be applied on a sample of possibly relevant documents, i.e. re-ranking, rather than the full document collection for efficiency reasons (Liu, 2009). Typically, an initial list of documents is retrieved for each query using an initial ranker then features are extracted from those documents only and used to train or apply the learning to rank model. The proposed framework expands on these two main steps by adding two intermediate components to match each search session to its related social positions and to identify related search sessions. Algorithm 5 presents an algorithmic summary of the proposed session-based system.

6.4.1 Initial ranker

Before describing the initial ranker, it is important to note that there are two search engines involved in my experiments. The first is a search engine used by the TREC session track organisers with which users interacted. I refer to this system as *the observer*. The second engine is the experimental system I used to index and retrieve from the document collections, referred to as *local*. Both systems use different retrieval models and vary in their indexing parameters. Thus, results for each query also vary. It is likely that the user might behave differently if interacting with the local system[6].

In ad-hoc retrieval tasks, it is common to use standard retrieval models (e.g. QL or BM25) as the initial ranker. In session search, there are multiple queries for each session and richer contextual information about the user. It is sensible to apply a custom initial ranker that would increase the effectiveness of the initial results using a session's context. Therefore, I developed a simple initial ranker based on query formulation methods, which

[6] For example, by submitting different reformulations or clicking on documents that were not retrieved by the other system.

Algorithm 5: An algorithmic summary of the proposed session-based search system.

```
Input:
$q$: a search query.
$s$: a current search session.
$M$: social positions' models.
Output:
$D_i$: an initial list of ranked documents for query $q$.
$M_s$: a set of social positions' models relevant to session $s$
$T_s$: a set of social expansion terms for session $s$.
$R_s$: a set of related search sessions relevant to session $s$.
$D_q$: a final list of ranked documents for query $q$.
begin
    $D_i \longleftarrow$ Retreive top $k$ documents for the query $q$ using equation (6.9).
    $M_s \longleftarrow$ Identify a subset of social positions' models $M$ that is relevant to the
      session $s$ using equation (6.12).
    $T_s \longleftarrow$ Extract a set of social expansion terms that is relevant to the session $s$
      and the session's social positions $M_s$ using equation (6.15).
    $R_s \longleftarrow$ Identify a set of related sessions from query logs using an AROW
      classifier based on features as in table (6.3).
    foreach $j = 1...K$ do
        $\boldsymbol{x}_{d_j,g} \longleftarrow$ Extract general learning to rank features for each $d_j \in D_i$.
          Features are listed in table (6.4).
        $\boldsymbol{x}_{d_j,s} \longleftarrow$ Extract social position dependent features based on $M_s$, $T_s$, and
          $R_s$ for each $d_j \in D_i$. Features are listed in table (6.5).
        $\boldsymbol{x}_{d_j} \longleftarrow$ Combine $\boldsymbol{x}_{d_j,s}$ and $\boldsymbol{x}_{d_j,g}$ into a single feature vector.
    end
    $D_q \longleftarrow$ Run a pre-trained learning to rank model $f(\boldsymbol{x})$ where $\boldsymbol{x} = \left\{\boldsymbol{x}_{d_j}\right\}_{j=1}^{k}$
end
```

have been shown to be effective in session search (Guan and Yang, 2014; Jiang and He, 2013; Jiang et al., 2012; Van Gysel et al., 2016). Formally, the initial ranker query model $\widehat{\theta}_q$ is estimated as follows:

$$P(w|\widehat{\theta}_q) = \alpha P_{MLE}(w|\theta_{concat}) + (1-\alpha)P(w|\phi_q^*) \tag{6.9}$$

θ_{concat} is a query model estimated over the concatenation of all queries in the session using maximum likelihood estimation. Van Gysel et al. (2016) found that concatenating all queries in a session led to improved performance. Let $\phi_i^{observer}$ and ϕ_i^{local} represent the sets of relevance models for session i by the observer and the local engines, respectively. I score each $\phi_{i,n}^{observer}$ based on the following function:

$$ModelScore(\phi_{i,n}^{observer}) = \underset{\phi \in \phi_i^{local}}{\arg\max}\, Jaccard(\phi_{i,n}^{observer}, \phi) \tag{6.10}$$

where $Jaccard(.,.)$ is the Jaccard coefficient between the two models' terms. ϕ_q^* is then selected as follows:

$$\phi_q^* = \underset{\phi \in \phi_i^{observer}}{\arg\max}\ ModelScore(\phi) \tag{6.11}$$

The goal is to select a set of expansion terms that both systems believe is relevant to the test session. The assumption here is motivated by Lee's (1997) hypothesis in data fusion research who claims that different retrieval models might return similar sets of relevant documents but not non-relevant documents. Similarly, if the two different systems produce a similar relevance model for a particular query in the test session, it is likely that this relevance model would be composed of expansion terms relevant to the sought-after session's information need and thus using such terms would likely improve the initial ranker performance.

6.4.2 Matching search sessions to social positions

Search queries, as discussed previously, are typically short. This empirical fact means that matching a search query to social positions is likely to be less accurate due to linguistic ambiguity. As a result, documents that are returned for a search query could be used as a surrogate representation of the query. Such a representation is then matched to social positions. In this section, I propose to extract terms from documents that are returned for any query of a current search session. This set of terms represents the current session and is subsequently used to match the session to social positions dynamically. In section 5.4.1, I proposed matching of a query to its most similar social positions using a static and query-independent representation of each social position. One limitation of the static approach is that a document d that is relevant to social position p might use a set of terms that are not well represented in p's model or p's document representation. In this situation, the similarity score between them would be low because of the vocabulary mismatch problem, possibly resulting in lower matching accuracy. To overcome this limitation, I rely on the concept of word embeddings to build a vector representation for each term in the test collection vocabulary.

This type of representation enables a similarity calculation to be made between a term t and social position p by averaging the similarity scores between t's vector and vectors for each term in p's model. A document or a search session can be represented as a set of terms T. The task of matching a search session to a social position is then cast as finding the social position that is most similar to the terms set T. The assumption made here is that terms relevant to social position p should be semantically similar to p's terms. An important by-product of this approach is that a set of expansion terms relevant to the search session and the matched social position can be jointly extracted.

Formally, let session i be represented by a tuple $S_i = \langle Q_i, D_i, \theta_i, fb_i \rangle$. The session's list of queries is represented by $Q_i = \{Q_{i,1}, Q_{i,2}, \ldots, Q_{i,n}\}$ where $Q_{i,n}$ is the session's current query to be personalised. $D_i = \{D_{i,1}, D_{i,2}, \ldots, D_{i,n}\}$ is the top $n = 10$ documents for each query in the session. $\phi_i = \{\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}\}$ is the set of relevance models for each iteration. These relevance models are computed using relevance modelling (Lavrenko and Croft, 2001). For instance, $\phi_{i,n}$ is computed over $D_{i,n}$ for query $Q_{i,n}$ in session i. $fb_i = \{fb_{i,1}, fb_{i,2}, \ldots, fb_{i,n}\}$ is the set of expansion, feedback, terms for each iteration. It contains the top $n = 40$ terms in the relevance model of each query. Note that there are different D_i for the observer and local systems and thus different ϕ_i and fb_i.

Let T_i^{observer} denote the set of relevant terms for session i using the observer system. $T_i^{\text{observer}} = \bigcup_{1 \leqslant j \leqslant n} fb_{i,j}$. A unified $T_i = T_i^{\text{observer}} \cup T_i^{\text{local}}$. In other words, T_i contains expansion terms that the observer or the local systems believe are relevant to session i. Furthermore, let R_j denote the set of terms representing the social position j. The similarity between social position j and search session i is then defined as follows:

$$Sim(T_i, R_j) = \frac{1}{|T_i|} \sum_{x \in T_i} \frac{1}{|R_j|} \sum_{z \in R_j} \frac{\mathbf{T_{i,x} R_{j,z}}}{\|\mathbf{T_{i,x}}\| \|\mathbf{R_{j,z}}\|} \tag{6.12}$$

where $T_{i,x}$ and $R_{j,z}$ are vector representations for terms $x \in T_i$ and $z \in R_j$. To obtain such a vector representation for each word, I trained a continuous bag of words (CBOW) model (Mikolov et al., 2013) using ClueWeb09B collection[7].

CBOW learns to predict a centre word based on its neighbours in a fixed window using two weight matrices: $\mathbf{W}$ of size $V \times N$ from the input layer to the hidden layer and $\mathbf{W}'$ of size $N \times V$ from the hidden layer to the output layer. V is the vocabulary size and N is the number of dimensions. Let u_j represent the score of the j^{th} word in a particular window. u_j is calculated as follows (Rong, 2014):

$$u_j = \mathbf{v}'^{T}_{w_j} \cdot \frac{1}{C}(\mathbf{v}_{w_1} + \mathbf{v}_{w_2} + \cdots + \mathbf{v}_{w_c}) \tag{6.13}$$

where C is the window size. $\mathbf{v}'_{w_j}$ is the j^{th} column in $\mathbf{W}'$. $\mathbf{v}_{w_c}$ is the c^{th} context word's row in $\mathbf{W}$. u_j is converted to a probability using the softmax function as follows:

$$P(w_j | w_1, w_2, \ldots, w_c) = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} \tag{6.14}$$

CBOW's training objective is to maximise the log probability of equation 6.14. After training, $\mathbf{W}$ is the weight matrix used to obtain word embeddings. After scoring all social positions using equation 6.12, I take the r most similar ones as session's i social positions.

The second objective of this section is to identify a set of terms that are relevant to both the session and the session's social positions. Let E_i denote this set such that $E_i \subseteq T_i$.

[7] Spam documents were excluded.

No.	Queries	Social positions	Expansion terms
T11.8	laser treatment definition → beauty or cosmetic and laser treatment	dermatologist, plastic surgeon	cellulite, about, resurface, scar, lasik, effective, skin, ipl, efficac, eye, define, safe, derm, botox, lipodissolve, wrinkle, safety, facial, cynosure, fraxel
T12.5	pocono mountains park → pocono mountains shopping	tour guide, tourist	accommodations, getaway, waterpark, attraction, honeymoon, camelback, vacation, trail, resort, hotel, trip, chalet, tour, lodging, lodge, picnic
T13.44	healthy meals → healthy nutrition for infants	weight watcher, food journalist	recipe, diet, nutritious, breakfast, fat, recipes, snack, calorie, seafood, food, serving, eat, lunch, gourmet
T14.44	cyprus economic crisis → european economic crisis	economic consultant, policy analyst	eu, issues, eurozone, international, economy, cepr, imf, monetary, foreign, governance

Table 6.2: Example sessions with their matched social positions and expansion terms. Tx.i refers to the i^{th} session in TREC session track x. For presentation reasons, only the last two queries of each session are presented.

Set E_i is referred to as social expansion terms. To populate E_i with terms from T_i, I first calculate the probability that term $x \in T_i$ belongs to social position j using the softmax function and equation 6.12

$$P(j|x) = \frac{\exp(Sim(x, R_j))}{\sum_{x' \in T_i} \exp(Sim(x', R_j))} \tag{6.15}$$

Let j' be the most probable social position for x. x will be added to E_i if j' is among the session's social positions. Table 6.2 presents example sessions, their matched social positions, and social expansion terms.

6.4.3 Identification of related search sessions

It is common to use query logs as a resource to study previous users' interactions with search engines. Findings derived from such studies help in improving diverse tasks. Three main findings are relevant to session search. Firstly, the distribution of search queries, and query terms, follows a power-law distribution which might be explained by the bursty nature of queries where multiple users initiate a search session about the same topic at close time intervals (Silvestri, 2010). Secondly, re-finding behaviour is common (Teevan et al., 2007a; Tyler and Teevan, 2010) where users look for new or already seen information about a topic that they have searched about before. Thirdly, not all information needs are satisfied by a single search session but some span across multiple sessions (Kotov et al., 2011). A manual analysis by Donato et al. (2010) revealed that approximately 10% of users' sessions are comprised of cross-sessions information needs and that such tasks are responsible for a quarter of query volume. All these empirical findings suggest that it is reasonable to assume that query logs might contain sessions that are similar to the current user's information need. The identification of such sessions forms the focus of this section.

Related sessions can be identified using term-based or content-based approaches. Luo et al. (2014a) constructed a term vector for each session from the combination of all queries in the session. Terms' idf values were assigned as weights. Vector representations for all sessions were then clustered using the k-means algorithm to discover topics in the query logs. Sessions belonging to the same cluster were considered related and used to boost the ranking of a document if it had been SAT-clicked in related sessions. Li et al. (2015) investigated the effectiveness of four classes of features with respect to: current query, query change, whole session, and related sessions. Features from the current query and global sessions were found to influence their LTR model more positively than the other two classes. To determine topically related sessions, they estimated an LDA topic model over clicked documents in the entire query log. They then built a topic vector for each session based on the session's clicked documents. The similarity between two sessions was calculated using the cosine similarity between their topic vectors.

In essence, this task is an online task that needs to be performed at query time. The method proposed by Li et al. (2015) requires an LDA topic model to be estimated over the entire clicked-documents in the query logs for each session, which is prohibitive in practice. The alternative of using term-based methods is relatively more efficient but inherits the standard issues of lexical similarity between search queries (e.g. two sessions might be about the same information need but use different vocabulary). In addition, both methods are based on clustering models that require the number of topics to be set a priori.

I have considered the task of identifying related sessions as a binary classification task with respect to a test session t. Formally, let S be the set of all sessions in the query logs. $x_{e,t}$ is a feature vector to represent the relatedness, or lack of, between test session t and $e \in S$. To train the classifier, I used the topic labels provided by the TREC Session search organisers. If two sessions have the same topic label, they are considered related. I use the AROW classifier as described in section 3.2.2. Table 6.3 presents the set of classification features. In its most basic implementation, a binary classifier will decide for each session in the query logs whether or not it relates to the test session t. However, query logs would typically contain billions of sessions and only a small fraction might be worth examining. Thus, pruning of unlikely candidate sessions is performed based on the following simple rule: related sessions must have at least one social position in common and at least one

Feature No.	Feature description
1	Number of shared query terms.
2	Ratio of shared query terms.
3	Number of identical queries in both sessions.
4	Number of shared results.
5	Ratio of shared results in both sessions.
6	Jensen-Shannon divergence between the two sessions' relevance model.
7	Jaccard similarity between the two sessions' expansion terms.
8	Jaccard similarity between the two sessions' social positions.
9	Jaccard similarity between the two sessions' social expansion terms.
10	Jensen-Shannon divergence between the two sessions' social relevance model.

Table 6.3: Features used to identify related sessions. For features 4 and 5, a cutoff of 10 results per query is applied.

shared result or query term. This pruning rule presupposes that for two sessions to be related, they must be relevant to at least one social position. This captures the semantic similarity between two candidate sessions without the need to run a computationally expensive topic modelling algorithm on the entire clicked documents demonstrated by Li et al. (2015). The identification of social positions is performed for each search session independently from any other sessions in the query logs as described in section 6.4.2

Features 1, 2 and 3 represent lexical similarity features while the remaining features can be considered semantic features. The last five features are computed using the output of the previous components, i.e. the initial ranker and the matcher. Features 8, 9 and 10 rely on identifying each session's social positions. Unlike the LDA-based approach of Li et al. (2015), this process is performed for each session separately. The session's relevance model is the same relevance model used in the initial ranker and is estimated using equation 6.11 in section 6.4.1. The session's social relevance model is a vector of social expansion terms weighted using each term's weight in the session's relevance model.

6.4.4 Learning to rank features

As mentioned in section 6.4, the approach presented in this chapter is a learning to rank one. In the previous sections, I described the initial ranking component used to generate a list of documents for each query to be re-ranked by the learning to rank model. The model I used is lambdaMART and a full discussion about learning to rank in general and lambdaMART in particular is provided in section 2.5.1. In this section, I describe the features that are used to represent candidate documents for each test query. In my experiments, there are two sets of features. The first is independent from the social positions of the test session whilst the second depends on identifying the session's social positions. Tables 6.4 and 6.5 list the features of both sets, respectively.

	Description	**Total**
First and current queries		
Document level.	First query's scores using QL, BM25 and HLM.	3
	Current query's scores using QL, BM25 and HLM.	3
	Average of first and current queries' scores using QL, BM25 and HLM.	3
Snippet level.	First query's scores using QL, BM25 and HLM.	3
	Current query's scores using QL, BM25 and HLM.	3
Aggregate query		
Document level.	Number of tokens in query.	1
	Number of distinct terms.	1
	Number of query terms in document.	1
	Query's terms ratio in document.	1
	Query's scores using QL, BM25 and HLM.	3
	Query model's scores using QL, BM25 and HLM.	3
Snippet level.	Query's scores using QL, BM25 and HLM.	3
	Query's terms ratio in snippet.	1
Aggregate query terms		
	Terms' statistics using QL. BM25, HLM.	15
	Top terms' scores using QL, BM25, HLM.	3
Session features		
	Number of queries in the session.	1
	Session's statistics using QL.	5
	Session's statistics using BM25.	5
	Session's statistics using HLM.	5
Expansion terms		
	Expansion terms' scores using QL, BM25 and HLM.	3
	Clicked documents' expansion terms using QL, BM25 and HLM.	3
	Rank using the initial ranker as in equation 6.9.	1
Document features		
	PageRank score (The Lemur project, 2009, 2012).	1
	Spamness score (Cormack et al., 2011).	1
	Stopwords ratio.	1
	Document length.	1
	Binary indicator for Wikipedia documents.	1
	Grand total	75

Table 6.4: Social position independent features.

In table 6.4, I consider five groups of features. The first four are query-dependent. These can be considered as different representations of the user's information need based on: the first and current queries, an aggregate query, expansion terms, and the list of session's queries. For each of those four groups, features are mostly based on scoring the relevance of the document, or the document's snippet, to the respective representation using the QL model (Ponte and Croft, 1998), BM25 (Robertson and Walker, 1994), and Hiemstra's language model (henceforth HLM) (Hiemstra, 1998).

In general, session search focuses on using a session's interaction data to personalise the results of the user's current query, i.e. the last query in the session. Therefore, it is intuitive to include features that represent the relevance of candidate documents to the current query. It is also logical to assume that some of the session's queries might capture the user's information need better than others. Guan and Yang (2014) investigated the question of which queries in session search are most important and should be assigned higher weights in an aggregation scheme. Besides the current query, they found that the first query is almost as important as the current query. My experiments on the initial ranker, see section 6.4.1, conform with Guan and Yang's conclusion. In most sessions, the first query provides the best relevance model and is selected more than any other query by the selection formula 6.11. I, therefore, include features to specifically account for the session's first query.

The second group is informed by previous research on query formulation, which focuses on composing a new query using the session's query chain. In particular, I include features representing two methods to build the new query. The first is based on Van Gysel et al.'s (2016) work where the new query is simply the concatenation of the query chain. This is called an aggregate query. The second method estimates a query model by interpolating a query model built over the session history H_Q with another query model θ_q that is built using the current query. The session history includes all queries prior to the current query. This is similar to the work by Jiang et al. (2012), as in equation 6.8, except that Jiang et al. included a relevance model in addition to the current and history models. Formally, the query model is built using the following equation:

$$P(w|\widehat{\theta_q}) = \lambda P_{MLE}(w|\theta_q) + (1 - \lambda) P_{MLE}(w|H_Q) \tag{6.16}$$

In addition, statistical relevance features for all the terms that the user used during the session are collected. These include: maximum, minimum, average, variance, and standard deviation. The top terms features are meant to represent the most frequent term or terms in the query chain. These are terms that the user insists on including the most during the session. The third group represents expansion terms, which are extracted using two approaches. The first is based on the relevance model selected using equation 6.11 as in the initial ranker component, section 6.4.1. For the second approach, I estimate a relevance model using relevance modelling (Lavrenko and Croft, 2001) over the session's clicked documents[8]. For both approaches, a cut-off of 40 terms is applied. I include the document rank based on the initial ranker component. The document rank is calculated as $\frac{1}{log_2(1+rank_d)}$, where $rank_d$ is the rank of document d using the initial ranker.

The session features measure the relevance of each candidate document to each query in

[8] All clicked documents are considered regardless of the dwell time.

Description	Total
Related sessions features	
Number and ratio of topic query's terms in document.	2
Topic query's scores using QL, BM25 and HLM.	3
Ratio of topic's expansion terms in document.	1
Topic relevance model scores using QL, BM25 and HLM.	3
Click relevance model scores using QL, BM25 and HLM.	3
Social position features	
Number and ratio of social expansion terms in document.	2
Scores of social expansion terms which appear in clicked documents titles using QL, BM25 and HLM.	3
Scores of social expansion terms which appear in clicked documents' snippets using QL, BM25 and HLM.	3
Ratio of the topic-level social expansion terms in document.	1
Social relevance model scores using QL, BM25 and HLM.	3
Scores of the topic-level social relevance model using QL, BM25 and HLM.	3
QL, BM25 and HLM scores of the topic-level social relevance model that is built using clicked documents.	3
Grand total	30

Table 6.5: Social position dependent features.

the session. This is approached by collecting the following statistical measures: maximum, minimum, average, variance, and standard deviation. The assumption is that a relevant document for the current query would likely be relevant to the previous queries as well with less variance if the session's information need is coherent enough. Finally, document features represent candidate document's quality features such as its PageRank score and spamness score. I also include a binary indicator for "Wikipedia" documents. Previous work by Lungely et al. (2011) suggested that by expanding the query with the term "Wikipedia" a better performance on session search is obtained. This might indicate that, for some sessions, entries from "Wikipedia", as a popular destination for searchers, are likely to be an authoritative resource (White et al., 2007).

Table 6.5 presents the second set of features. They are organised into two groups: current session and related sessions. The current session features depend on identifying the social position of the current session as described in section 6.4.2. For the second group, related sessions are identified as discussed in section 6.4.3, which utilises social position data to group related sessions. It is important to note that related sessions could be identified using other methods that do not require the identification of a session's social position. For example, Li et al. (2015) used an LDA-based method, discussed in section 6.4.3, for this purpose. I developed a baseline based on Li et al.'s method.

In section 6.4.3, I discussed the design of a classifier to identify related sessions. The classifier identifies all related sessions to a test session. Relationships between such identified sessions and the test session are assumed to be transitive. Thus, the results of classifying all sessions in the query logs is a set of hard clusters, i.e. mutually exclusive. Each cluster is called a topic. The related sessions features measure the relevance of candidate documents to four novel representations: topic query, topic relevance model, topic expansion terms, and topic-clicked relevance model. The topic query is the concatenation of all related sessions' queries and the current session's queries. An inherent assumption is that the method by which related sessions are identified is fine-grained so that related sessions are about the same information need and not broadly related. Thus, the topic query would likely contain repeated terms representing the key term, or theme terms, of such an information need. The topic relevance model θ_t is computed as follows:

$$P(w|\theta_t) = \frac{\sum_{s \in R} P(w|\theta_s)}{|R|} \tag{6.17}$$

where R is the set of related sessions and θ_s is the relevance model for session s as selected using equation 6.11 in section 6.4.1. Topic expansion terms are defined as the top 40 terms in the topic relevance model θ_t. The topic-clicked relevance model is estimated using relevance modelling (Lavrenko and Croft, 2001) over all clicked documents in sessions that belong to the same topic cluster as the current session.

In section 6.4.2, I described a method for matching the current session i to its most relevant social positions. One objective of this matching process was to identify a set of terms E_i that is likely to be relevant to the current session and its social position. This set is called social expansion terms and examples are shown in table 6.2. A further three sets of social expansion terms are derived and used to introduce features in table 6.5. The first is a subset of E_i containing terms that occur in the current session clicked documents' titles and another one for terms appearing in the snippets of clicked documents. The assumption is that these two sets would contain highly relevant terms to both the session and its social positions that possibly triggering the user to click on these documents. The third set contains topic-level social expansion terms built as the union of all the social expansion terms for sessions that belong to the same topic R as $\bigcup_{s \in R} E_s$.

Social position features also depend on three types of relevance models. The first is the social relevance model. For session i, this model's terms are the social expansion terms E_i. Terms are weighted based on the session's relevance model for one of the session's queries as selected by equation 6.11. The weighted average of all social relevance models for sessions that are members of the same topic forms a topic-level social relevance model. It is constructed in a similar way as the topic relevance model in equation 6.17 except that its components must be in $E_i \, \forall \, i \in R$. Finally, a third variation is estimated using clicked documents in sessions that are related to the current session. The components of this model are also limited to social expansion terms only.

6.5 Experimental setup

The main hypothesis of this chapter is: *the relevance of search results for a test query can be effectively and significantly improved by using user models of the session's social positions.* To investigate this hypothesis, I aim to answer the following research questions:

RQ1. How effective is the proposed learning to rank approach for session search compared with other well-established systems?

RQ2. What is the significance of features estimated using social positions' models?

RQ3. Does the identification and use of related sessions' data improve performance and how effective is the role-based user modelling framework in identifying related sessions compared with an alternative approach?

RQ4. Which sessions are better personalised than others using the proposed approach?

In this section, I describe the experimental settings while results are discussed in section 6.6.

6.5.1 Test queries and retrieval collection

I evaluated the proposed approach on TREC11-2014 session tracks (Carterette et al., 2013, 2014; Kanoulas et al., 2012, 2011b). There were $1,282$ test queries in total. A detailed description of this test collection is provided in section 6.3.1. The session tracks of 2011 and 2012 used ClueWeb09 as their document collection and ClueWeb12 for the 2013 and 2014 tracks. I used category B from both collections in my experiments, composed of approximately about 50 million pages per collection. In addition, the organisers of the TREC14 session track provided a baseline run[9] to each participant to use. I indexed all of the documents that were included in the baseline run since some of them are not included in category B of ClueWeb12. All experiments on TREC14 are based on the organisers' baseline run. The retrieval system used was the same one I developed for the diversification experiments in chapter 5, with similar pre-processing steps, i.e. stemming using the Krovetz stemmer (Krovetz, 1993) and stopwords removal.

6.5.2 Evaluation procedure

The approach presented in this chapter, as discussed previously, is a re-ranking approach based on an initial list of candidate documents. This list was retrieved using the initial ranker, as in section 6.4.1. Documents with a spam percentile of less than 70 were removed

[9] Available at `http://ir.cis.udel.edu/sessions/2014baseline.RL1.gz`

(Cormack et al., 2011). This list of candidate documents was truncated at rank 50 for all queries and were then used by the learning to rank model to produce the final runs. The evaluation metrics used in this chapter are based on TREC session track's official metrics which are: nDCG@k in equation 2.32, nERR@k in equation 2.29, and MAP. All runs are evaluated using the official evaluation script and qrels. Statistical tests are performed using paired t-test with Bonferroni correction ($p < 0.05$).

To validate my proposed approach, I compared with the following systems in addition to the initial ranker:

- **Current query:** A retrieval system based on the QL model (Ponte and Croft, 1998) with a Dirichlet smoothing parameter $\mu = 3500$. This system uses the current query only.

- **Best TREC:** This baseline refers to the best performing runs for each TREC Session track. These are: wildcat2 for 2011 (Kanoulas et al., 2011b), PITTSHQM for 2012 (Jiang et al., 2012), FixInt28 for 2013 (Jiang and He, 2013), GUS14Run3 for 2014 (Luo et al., 2014a). Both PITTSHQM and FixInt28 are query formulation methods that estimate a new query model based on variations of equation 6.8 and equation 6.4, respectively. The GUS14Run3 is a QCM-based system and is thus an MDP approach.

- **Aggregated query:** A concatenation of all of the session's queries as suggested by Van Gysel et al. (2016). The retrieval model is QL with similar settings as in the current query baseline.

- **QCM:** The Query Change Model with default parameters as used by Guan et al. (2013). I discussed QCM in section 6.2.3.

- **LTR:** There are three configurations of my proposed approach. **LTR-Base** uses the features in table 6.4 only and does not take advantage of the social positions dependent features in table 6.5. Thus, it serves as a learning to rank baseline. The second configuration is called **LTR-LDA**. The features used to train this baseline are displayed in table 6.4 and the related sessions features in table 6.5. This system uses LDA to identify topically related sessions as done by Li et al. (2015). The LDA topic model was estimated based on 1000 iterations with default hyper-parameter values (Griffiths and Steyvers, 2004) and the number of topics was set to 100. **LTR-SP** is the main approach of this chapter. It uses the full list of features in table 6.4 and table 6.5. Related sessions are identified using the social positions based classifier that was presented in section 6.4.3.

In terms of parameters, a trade-off parameter α is used to interpolate between the concatenated query model θ_{concat} and the best query's relevance model ϕ_q^* in the initial ranker equation 6.9. A similar parameter λ is used to interpolate the current query model θ_q with a history query model H_q in equation 6.16. The resultant query model is used as a feature for the learning to rank model. Both parameters were set to an equal value ($\alpha = \lambda = 0.70$). To train the learning to rank models, I performed a 10-fold cross validation by splitting the queries into training (60%), validation (20%), and test (20%) sets. I used the lambdaMART implementation in the RankLib[10] library with default parameters (number of trees=1000, leafs=10, learning rate=0.1). Statistical significance tests are performed in comparison to: initial ranker and QCM. I used the Gini impurity (Breiman et al., 1984; Shih, 1999) to calculate the importance of learning features. The Gini impurity for a node p is calculated as follows (Shih, 1999):

$$gini(p) = 1 - \sum_c p_c^2 \tag{6.18}$$

where p_c is the relative proportion of label c in node p. The importance of node p is calculated as follows:

$$importance(p) = w(p)gini(p) - w(p_l)gini(p_l) - w(p_r)gini(p_r) \tag{6.19}$$

where p_l and p_r are the left and right splits on node p, respectively. $w(p)$ is the weighted number of instances reaching node p.

6.6 Results

In the following sections, I discuss the results and contributions of the main four components of my approach. In section 6.6.1, I investigate the effectiveness of the initial ranker then section 6.6.2 focuses on RQ1 by validating the proposed learning to rank approach using four public test collections. Section 6.6.3 answers RQ2 by analysing the contribution of social positions' models to the performance of the proposed approach. Section 6.6.4 focuses on RQ3 by evaluating the role of related sessions' features in improving session search. This section includes a comparative analysis of two approaches to identifying related sessions. The first is a novel method based on social positions, as presented in section 6.4.3, and the second is an LDA-based algorithm (Li et al., 2015). Lastly, section 6.6.5 investigates RQ4 by analysing the performance of the proposed approach on various session types.

6.6.1 Initial ranker

Tables 6.6 and 6.7 show the results of the proposed initial ranker compared with the current query, the aggregated query, and two different instantiations of the initial ranker.

[10] https://sourceforge.net/p/lemur/wiki/RankLib/

	TREC 2011		TREC 2012	
	nDCG@10	nERR@10	nDCG@10	nERR@10
Current	0.3480	0.3968	0.2478	0.2991
Aggregated	0.4066 (16.84%)	0.4644 (17.04%)	0.2941$^{\uparrow}$(18.68%)	0.3449 (15.31%)
Initial (select)	**0.4427**$^{\uparrow\bullet}$(27.21%)	**0.4980**$^{\uparrow\bullet}$(25.50%)	**0.3464**$^{\uparrow\bullet}$(39.79%)	**0.3899**$^{\uparrow\bullet}$(30.36%)
Initial (first)	0.4300$^{\uparrow}$(23.56%)	0.4844$^{\uparrow}$(22.08%)	0.3420$^{\uparrow\bullet}$(38.01%)	0.3859$^{\uparrow\bullet}$(29.02%)
Initial (last)	0.4238$^{\uparrow}$(21.78%)	0.4837$^{\uparrow}$(21.90%)	0.3411$^{\uparrow\bullet}$(37.65%)	0.3882$^{\uparrow\bullet}$(29.79%)

Table 6.6: Performance of different initial ranker's configurations on TREC 2011 and 2012.

	TREC 2013		TREC 2014	
	nDCG@10	nERR@10	nDCG@10	nERR@10
Current	0.1000	0.1337	0.1937	0.2263
Aggregated	0.1302$^{\uparrow}$(30.20%)	**0.2031**$^{\uparrow}$(51.91%)	0.2028 (4.70%)	0.2489 (9.99%)
Initial (select)	**0.1303** (30.30%)	0.2010$^{\uparrow}$(50.34%)	**0.2125** (9.70%)	0.2596 (14.71%)
Initial (first)	0.1254 (25.40%)	0.1928$^{\uparrow}$(44.20%)	0.2107 (8.77%)	**0.2638** (16.57%)
Initial (last)	0.1271 (27.10%)	0.1967$^{\uparrow}$(47.12%)	0.2049 (5.78%)	0.2535 (12.02%)

Table 6.7: Performance of different initial ranker's configurations on TREC 2013 and 2014.

As described in section 6.4.1, the initial ranker interpolates two query models. The first is based on an aggregated representation of session's queries and the second is based on a relevance model estimated using a selected query based on equation 6.11. The initial *first* uses the relevance model of the first query in the session while the initial *last* uses the last query. Note that the last query is defined as the query before the current with which a user has been presented with results. Statistically significant improvement over the current query and the aggregated query are denoted using the symbols ($^{\uparrow}$) and ($^{\bullet}$), respectively. Results are reported using nDCG@10 and nERR@10. The percentage changes from the current query scores are also shown in the tables.

Table 6.6 shows that the proposed initial ranker model, Initial-Select, provided the best performance over all other systems for TREC 2011 and TREC 2012 test collections. The performance was statistically significant over the current query in terms of both nDCG@10 and nERR@10 with an improvement by 27.21% on nDCG@10 and 25.50% on nERR@10 for TREC 2011. Statistically significant improvements were also achieved on TREC 2012 for both metrics with an improvement over the current query by 39.79% and 30.36% using nDCG@10 and nERR@10, respectively. Initial-Select also provided significant improvements over the aggregated query using these two test collections for both evaluation metrics. Table 6.7 shows that Initial-Select still provides the best performance in terms of nDCG@10 for TREC 2013 and TREC 2014 collections. However, it did not obtain the best scores in terms of nERR@10 nor provided any statistically significant difference in terms of nDCG@10 compared with the current or aggregated queries. In

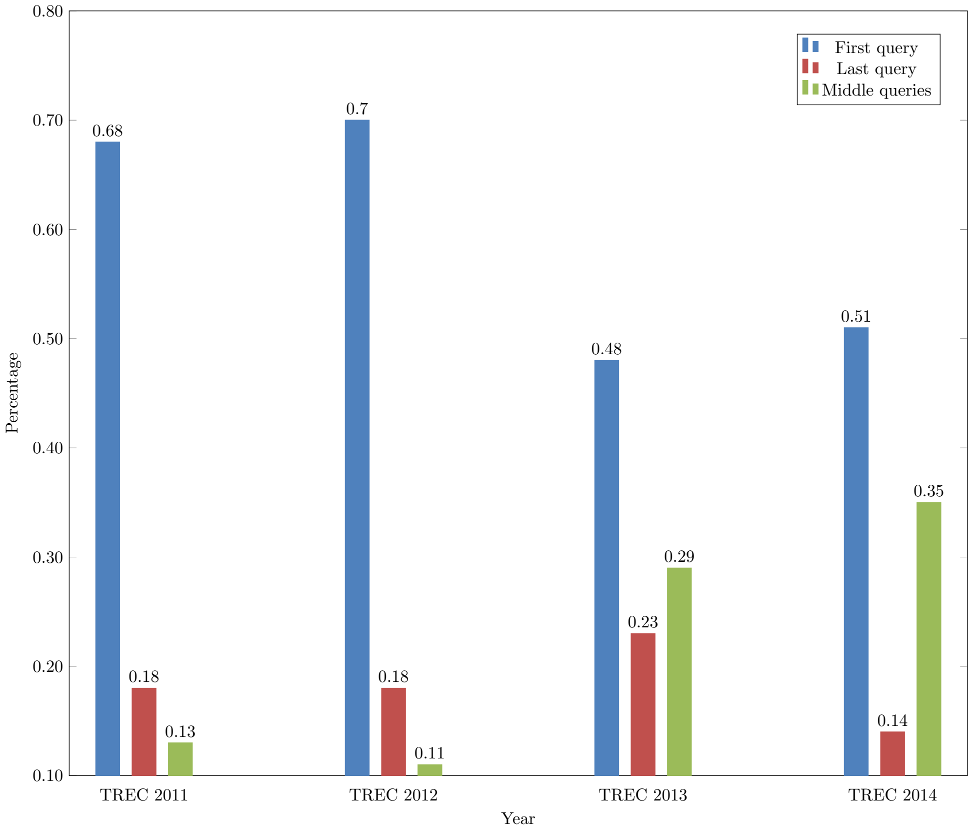


Figure 6.2: Distribution of selected queries for the initial ranker per test collection.

general, the performance of Initial-Select is consistently outperforming all other methods based on nDCG@10.

One main observation from tables 6.6 and 6.7 is that all methods provided improvements over the current query approach. This indicates that the incorporation of sessions' interaction data helps in improving the results for the current query. This was evident from the change percentage each method provided over the current query system. Although the patterns for these change percentages are comparable for TREC 2011, TREC 2012 and TREC 2013, they are lowest for TREC 2014. This is perhaps due to the use of the baseline run that was provided by the task organisers. For TREC 2014, all systems in the above tables re-rank this baseline run as opposed to retrieving results from the document collection. This was done because most of the relevant documents were in category A of ClueWeb12, which was not indexed.

The Initial-Select approach selects a relevance model estimated using results of one of the session's queries as in equation 6.11. As mentioned previously, the task organisers used a different retrieval system to collect the interaction data than the local system, which is used to retrieve results for evaluation. The main assumption behind the Initial-Select was that both systems would agree on a set of relevant expansion terms but not non-relevant. Thus, the relevance model that contained most shared terms is the best to use. As shown

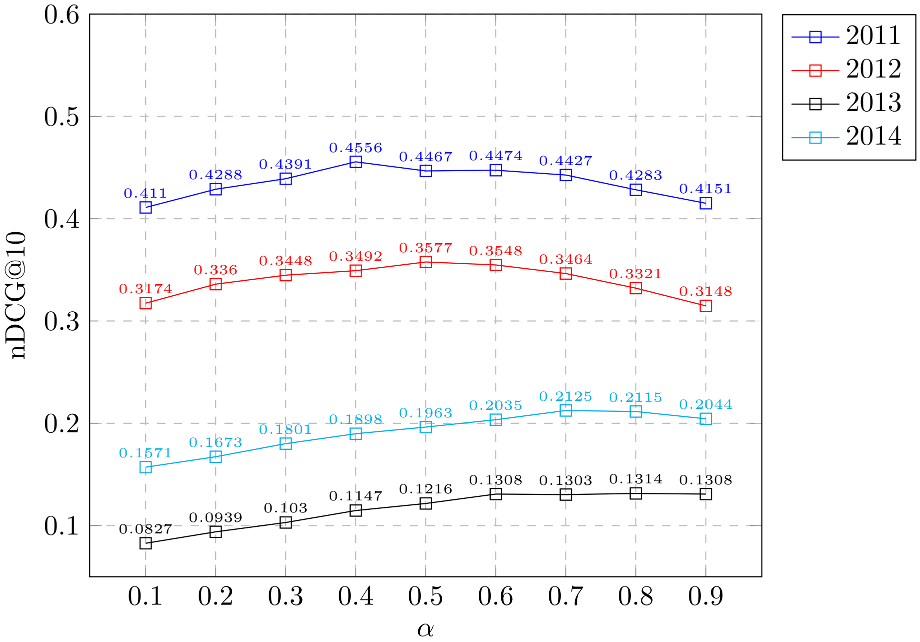


Figure 6.3: Performance of the initial ranker using different values of the interpolation parameter α on nDCG@10 for TREC 2011, 2012, 2013 and 2014.

in figure 6.2, the first query seemed to be predominantly selected as the best query to estimate the relevance model. However, if the first query is always selected (i.e. Initial-First) performance would be second to that of the Initial-Select. In fact, Initial-Select provided an average improvement of about 2.25% over Initial-First on nDCG@10. This result might support the Guan and Yang (2014) finding that the first query is of special importance. One possible explanation of such a behaviour is that users might tend to start their session with a general query that contains the key terms with none or few modifiers, i.e. a top-down search approach (Navarro-Prieto et al., 1999).

Figure 6.3 plots the performance of Initial-Select with different values for the interpolation parameter α. A low value represents a higher weight for the relevance model compared with the aggregated query. This figure shows two different trends. In the first, optimal performance for TREC 2011 and 2012 can be obtained at $\alpha = 0.40$ and $\alpha = 0.50$, respectively. This indicates almost equal importance for both the aggregated query model and the relevance model, although the latter seems to be more effective than the aggregated query for TREC 2011. In contrast, the optimal weighting value shifts in favour of the aggregated query at $\alpha = 0.80$ and $\alpha = 0.70$ for TREC 2013 and 2014. In fact, one can observe two different trends that separate TREC 2011 and 2012 from TREC 2013 and 2014 in figure 6.2 as well as tables 6.6 and 6.7. In figure 6.2, the first query is selected almost half the time for TREC 2013 and 2014 compared with about two thirds for TREC 2011 and 2012. The last query was the second best option for TREC 2011 and 2012 whereas it was one of the middle queries for the other two test collections. Initial-Select provides a significant improvement over the aggregated query for TREC 2011 and 2012, as shown in tables 6.6 and 6.7. Only a minor improvement, however, was observed for TREC 2013 and 2014. These trends could be attributed to the choice of the retrieval system used to collect users' interactions. For TREC 2011 and 2012, sessions' data were collected using the Yahoo! BOSS (Build your Own Search Service). A weighted query using an Indri-based system was used for TREC 2013 and 2014. The Yahoo! BOSS is expected to provide results with higher relevance and therefore the selected relevance model would be more effective leading to a statistically significant improvement over the aggregated query. It would also explain the higher trust that is placed on the relevance model for TREC 2011 and 2012, i.e. α value. In contrast, the Indri-based configuration is perhaps not greatly different from my local system. This probably shows a dependence on the quality of the retrieval system based on which the relevance model is estimated.

6.6.2 Approach validation

In this section, I address the research question RQ1. To validate the effectiveness of my proposed approach LTR-SP, I compare its performance to other related approaches, including state-of-the-art systems. Results using the TREC 2011 and 2012 test collections are presented in table 6.8 while table 6.9 reports results on TREC 2013 and 2014. Statistically significant improvements over the initial ranker and the QCM system are denoted with the symbols ($^\bullet$) and ($^\uparrow$), respectively. The change percentages and the number of affected queries (+ positive, - negative, = no changes) compared with the current query model are reported for all runs. The best scores are highlighted in bold for each test collection.

As shown in tables 6.8 and 6.9, LTR-SP performed substantially better than all other systems in terms of nDCG@10 and nERR@10. LTR-SP also provided the best MAP scores for all the test collections except for TREC 2013, although it is not optimised to improve MAP. LTR-SP improvements over the QCM and the initial ranker baselines were statistically significant with regard to nDCG@10 and nERR@10 on all datasets. Furthermore, LTR-SP significantly outperformed QCM on MAP for all test collections. The change percentages across all years also provided further support for LTR-SP. For instance, LTR-SP improvements relative to the best TREC systems were 10.26%, 21.30%, 10.96%, and 24.88% in terms of nDCG@10 for TREC 2011, 2012, 2013, and 2014, respectively. Under nERR@10, LTR-SP improvements over the best TREC system were by 9.41%, 32.07%, 37.48%, and 26.84% on TREC 2011 to 2014. This trend was also apparent when LTR-SP was compared with the other approaches for both metrics nDCG@10 and nERR@10 and for MAP except on TREC 2013. For TREC 2013, the most relevant documents originated from ClueWeb12 category A rather than its subset category B that is used by LTR-SP. This, perhaps, explains the difference in terms of MAP between the best TREC system that uses category A and LTR-SP.

	nDCG@k					nERR@k					MAP
	k=10	%	+	=	-	k=10	%	+	=	-	
TREC 2011:											
current	0.3480					0.3968					0.0824
Aggregated	0.4066	16.84	33	22	21	0.4644	17.04	32	22	22	0.1031
Initial ranker	0.4427$^{\uparrow}$	27.21	39	23	14	0.4980$^{\uparrow}$	25.50	39	23	14	0.1097
Best TREC	0.4540	30.46	43	16	17	0.5208$^{\uparrow}$	31.25	43	16	17	0.1253
QCM	0.4079	17.21	30	24	22	0.4550	14.67	29	24	23	0.1130$^{\bullet}$
LTR-Base	0.4638$^{\uparrow}$	33.28	44	16	16	0.5195$^{\uparrow}$	30.92	45	16	15	0.1291$^{\uparrow}$
LTR-LDA	0.4646$^{\uparrow}$	33.51	44	17	15	0.5231$^{\uparrow}$	31.83	41	17	18	0.1303$^{\uparrow}$
LTR-SP	**0.5006**$^{\uparrow\bullet}$	43.85	47	15	14	**0.5698**$^{\uparrow\bullet}$	43.60	48	15	13	**0.1335**$^{\uparrow}$
TREC 2012:											
current	0.2478					0.2991					0.1183
Aggregated	0.2941	18.68	39	31	28	0.3449	15.31	39	31	28	0.1387
Initial ranker	0.3464$^{\uparrow}$	39.79	58	26	14	0.3899	30.36	55	26	17	0.1576$^{\uparrow}$
Best TREC	0.3221	29.98	48	29	21	0.3595	20.19	46	29	23	0.1457$^{\uparrow}$
QCM	0.2746	10.82	41	23	34	0.3218	7.59	40	23	35	0.1169
LTR-Base	0.3712$^{\uparrow}$	49.80	58	25	15	0.4420$^{\uparrow\bullet}$	47.78	56	25	17	0.1541$^{\uparrow}$
LTR-LDA	0.3823$^{\uparrow\bullet}$	54.28	58	23	17	0.4645$^{\uparrow\bullet}$	55.30	60	23	15	0.1587$^{\uparrow}$
LTR-SP	**0.3907**$^{\uparrow\bullet}$	57.67	58	24	16	**0.4748**$^{\uparrow\bullet}$	58.74	61	24	13	**0.1620**$^{\uparrow}$

Table 6.8: Search accuracy on TREC Session tracks of 2011 and 2012.

Besides the average gain provided by LTR-SP under the considered evaluation metrics, it is important to analyse the robustness of LTR-SP based on the volume of queries that are positively and negatively affected by this approach. A robust approach would not hurt many queries. Tables 6.8 and 6.9 display the number of queries affected by each approach compared with the current query system, i.e. not using any session's data. Again, LTR-SP is superior to the other approaches. LTR-SP improved the performance of approximately 64% of all the test queries in the tables. This was the highest percentage of positively affected queries. When considering non-LTR approaches, it is followed by the best TREC system at 54% and QCM at 50%. In terms of hurt queries, LTR-SP had the lowest percentage at 19% compared with the best TREC system at 28%, the initial ranker at 28% and the QCM at 30%. For LTR-based systems, LTR-SP also had the highest number of helped queries (233) and the lowest hurt queries (68). These statistics are in terms of

	nDCG@k					**nERR@k**					**MAP**
	k=10	%	+	=	-	k=10	%	+	=	-	
TREC 2013:											
current	0.1000					0.1337					0.0322
Aggregated	0.1302	30.20	39	21	27	0.2031	51.91	43	21	23	0.0443
Initial ranker	0.1303	30.30	41	15	31	0.2010	50.34	46	15	26	0.0448
Best TREC	0.1706	70.60	49	10	28	0.2049	53.25	47	10	30	**0.0873**$^{\uparrow\bullet}$
QCM	0.1480	48.00	53	18	16	0.2108	57.67	50	18	19	0.0436
LTR-Base	0.1498	49.80	48	16	23	0.2259	68.96	49	16	22	0.0523$^{\bullet}$
LTR-LDA	0.1625$^{\bullet}$	62.50	49	15	23	0.2413$^{\bullet}$	80.48	48	15	24	0.0533$^{\bullet}$
LTR-SP	**0.1893**$^{\uparrow\bullet}$	89.30	56	11	20	**0.2817**$^{\uparrow\bullet}$	110.70	57	11	19	0.0593$^{\uparrow\bullet}$
TREC 2014:											
current	0.1937					0.2263					0.0819
Aggregated	0.2028	4.70	38	18	44	0.2489	9.99	40	18	42	0.0827
Initial ranker	0.2125	9.71	37	19	44	0.2596	14.71	37	19	44	0.0846
Best TREC	0.2580	33.20	56	7	37	0.3268$^{\bullet}$	44.41	59	7	34	0.0730
QCM	0.2443	26.12	57	8	35	0.3126	38.14	57	8	35	0.0636
LTR-Base	0.2934$^{\bullet}$	51.47	66	10	24	0.3694$^{\bullet}$	63.23	67	10	23	0.0931$^{\uparrow}$
LTR-LDA	0.3114$^{\uparrow\bullet}$	60.76	68	9	23	0.3913$^{\uparrow\bullet}$	72.91	70	9	21	0.0957$^{\uparrow}$
LTR-SP	**0.3222**$^{\uparrow\bullet}$	66.34	72	10	18	**0.4145**$^{\uparrow\bullet}$	83.16	74	10	16	**0.1055**$^{\uparrow}$

Table 6.9: Search accuracy on TREC Session tracks of 2013 and 2014.

nDCG@10. A similar observation was also noted for nERR@10. A breakdown of these numbers per year is shown in tables 6.8 and 6.9. Overall, the consistency of LTR-SP strong performance on these four test collections and its statistically significant improvement over the baselines under nDCG@10 and nERR@10 provide sufficient evidence that LTR-SP is an effective session search system. These results answer the first research question RQ1 regarding the effectiveness of LTR-SP compared with other session search systems.

6.6.3 Features analysis

In response to my second research question, I performed an ablation study to investigate the significance of features that are estimated using social positions' models. LTR-Base is a learning to rank model that is trained using general features which do not depend on inferring the session's social position. These features are listed in table 6.4. This

	nDCG@10	**nERR@10**	**MAP**
QCM	0.2638	0.3205	0.0836
Best TREC	0.2956	0.3471	0.1072
LTR-Base	0.3158	0.3861	0.1073
LTR-LDA	0.3270$^{\uparrow}$	0.4028$^{\uparrow}$	0.1099
LTR-SP	**0.3463**$^{\uparrow\bullet}$	**0.4316**$^{\uparrow\bullet}$	**0.1156**$^{\uparrow\bullet}$

Table 6.10: Performance of learning to rank approaches on 361 test sessions from TREC session tracks 2011-2014. Significant improvement over LTR-Base is denoted with ($^{\uparrow}$) while ($^{\bullet}$) indicates significant improvement over LTR-LDA.

	LTR-SP		**LTR-Base**	
	help	hurt	help	hurt
TREC 2011	+0.2857	-0.1305	+0.2653	-0.1792
TREC 2012	+0.2678	-0.0956	+0.2403	-0.1235
TREC 2013	+0.1633	-0.0600	+0.1458	-0.1160
TREC 2014	+0.2235	-0.1802	+0.2088	-0.1589

Table 6.11: Average difference in terms of nDCG@10 from the current query system.

system is compared with LTR-SP, which uses the same features as LTR-Base in addition to social position dependent features as in table 6.5. The results of both systems on the four test collections are shown in tables 6.8 and 6.9. Table 6.10 shows their performance and LTR-LDA on the 361 test queries from all four test collections. As can be noted from these tables, LTR-SP effectively outperformed LTR-Base on all test collections under the three evaluation metrics. Including social position's features increased the performance by an average of 12% on nDCG@10, 13% on nERR@10, and 9% on MAP.

The number of positively and negatively affected queries by both approaches seems comparable, although LTR-SP helped more queries and hurt fewer queries than LTR-Base. Table 6.11 reports the average difference for helped and hurt queries on all test collections for LTR-SP and LTR-Base. This shows that LTR-SP seemed to have the desired behaviour of lessening the negative difference compared with the current query system whilst increasing the positive one. It is also important to note that the performance of LTR-Base was better than all the other baselines, including the best TREC systems, for all test collections except TREC 2013. This, perhaps, indicated the usefulness of the general features in table 6.4. It also suggests that a learning to rank approach could be a viable solution for session search.

To further understand the relative contribution of features estimated using social positions' models, I calculated the Gini impurity (Breiman et al., 1984; Shih, 1999) for all trees averaged over all 10 cross-validation splits. These scores were then normalised relative to the feature with the highest Gini importance. Table 6.12 lists the top 25 features with those features that use social positions' models highlighted in bold. Approximately 48%

Feature	**Gini importance**
1. Stopwords ratio.	1.000
2. Rank using the initial ranker as in equation 6.9.	0.430
3. Topic query's score using BM25.	0.301
4. Spamness score (Cormack et al., 2011).	0.296
5. Topic relevance model's score using BM25.	0.227
6. Click relevance model's score using BM25.	0.226
7. Topic query's score using QL.	0.154
8. Aggregate query's scores using BM25 against document's snippet.	0.144
9. BM25's score of social expansion terms which appear in clicked documents' snippets.	0.135
10. Maximum BM25 score for aggregate query's terms.	0.123
11. Expansion terms' score using BM25.	0.104
12. First query's score using BM25 against document's snippet.	0.080
13. Current query's score using BM25 against document's snippet.	0.069
14. BM25's score of social expansion terms which appear in clicked documents' title.	0.068
15. HLM's score of the topic-level social relevance model which is built using clicked documents.	0.041
16. Maximum BM25 score for session's queries.	0.040
17. Average BM25 score for aggregate query's terms.	0.038
18. Ratio of topic's expansion terms in document.	0.035
19. HLM's score of social expansion terms which appear in clicked documents' title.	0.034
20. Average of first and current queries' scores using QL.	0.033
21. First query's score using QL.	0.031
22. Ratio of the topic-level social expansion terms in document.	0.028
23. Score of the topic-level social relevance model using BM25.	0.028
24. QL score of the topic-level social relevance model that is built using clicked documents.	0.026
25. Maximum QL score for session's queries.	0.024

Table 6.12: The relative importance of features based on their Gini scores. Features that use social positions' models are highlighted in bold.

of these top features were social position dependent and 52% general features. This is a close division between the two sets of features. An examination of this list, however, reveals the important role of social positions' features. Firstly, around 40% of all social positions' features were placed in the top 25 features compared with only 17% general ones. Secondly, there are 12 types of social positions' features as shown in table 6.5. 9 out of those 12 have at least one feature at the top of the list. This analysis and the ablation study demonstrates the significance of social positions' features and answers the second research question RQ2.

The significance of social positions' features is twofold. Firstly, social positions' models play a central role in efficiently and effectively identifying related sessions that are likely to be issued by a user with a similar information need. Session's social positions provide features for the classifier that identifies related sessions, as explained in section 6.4.3, as well as a rule for the early pruning strategy. The third and seventh important features in table 6.12 are about the topic query which, as mentioned earlier, is a concatenation of all related sessions' queries. Concatenating unrelated queries would likely harm rather than improve the results' relevance. Thus, it is critical to identify only closely related sessions. Secondly, in section 6.4.2, I introduced a method to extract social expansion terms whilst ensuring their relevance to both the session and its social positions. These terms proved to be useful especially if they appeared in the titles or snippets of clicked documents as in features 9, 14, 18, 19, and 22 in table 6.12. In addition, I used social expansion terms in building a social relevance model for the session and a variant of such a model estimated using social relevance models for all related sessions. Three out of the top 25 features were based on the topic-level social relevance model.

Surprisingly, the most important feature was the stopwords ratio. However, the presence of the spamness score at the fourth rank might give an indication of the importance of document's quality features. A high quality document is expected to contain a moderate number of stop-words and be less spammy. Stopword-based features have been shown to be effective in quality-biased ranking (Bendersky et al., 2011a). The distribution of general features at the top 25 features list is, however, to be expected. The first and current queries' group has 4 features in that list which, perhaps, enforces the early finding about the importance of the first query in session search. The other general groups with features in the top list are: aggregate query (3), session features (2), and expansion terms (1). These general features are key to the success of LTR-SP.

6.6.4 The identification of related sesssions

The feature analysis study provided a partial answer to the third research question RQ3. Features that were extracted from related sessions were shown to be among the most useful features. This is in line with Li et al. (2015) who reported a similar conclusion. Note that LTR-SP uses social position models to identify related sessions as in section 6.4.3. However, related sessions could be identified using other methods such as the LDA-based approach used by Li et al. (2015). To explore the effect of this group of features and the use of an alternative method of related sessions identification, I included the results of LTR-LDA in tables 6.8 and 6.9. LTR-LDA uses the related sessions' features in table 6.5 in addition to all the general features in table 6.4. Related sessions were identified using the LDA-based approach discussed in section 6.2.3.

If we consider all test sessions, displayed in table 6.10, across the four test collections, the nDCG@10 for LTR-LDA was 0.3270 compared with 0.3158 for LTR-Base. This showed an improvement of about 3.55% as a result of including related sessions' features. Similar trends were also observed in terms of nERR@10 and MAP. nERR@10 was 0.3861 for LTR-Base and 0.4028 for LTR-LDA (an improvement of about 4.33%). The MAP score for LTR-Base was 0.1073 and for LTR-LDA was 0.1099 (2.42% improvement). These improvements support the claim that related sessions' features are useful. The performance of LTR-SP was still superior to that of the LTR-LDA. Firstly, LTR-SP achieved 0.3463 on nDCG@10, 0.4316 on nERR@10, and 0.1156 on MAP. These were higher than LTR-LDA by 5.90%, 7.15%, and 5.19%, respectively. While LTR-SP has additional social position specific features, it identifies related sessions more efficiently than LTR-LDA. The latter requires estimation of an LDA model for all clicked documents in the query logs. This is an expensive operation to be performed at query time. In contrast, LTR-SP requires identification of each session's social positions separately. Then, the identification of related sessions was formulated as a binary classification task with few informative features that can be done online with minimal overhead.

Each test session is manually mapped to a specific topic by the TREC Session search organisers. Sessions belonging to the same topic can be considered as related. Thus, the performance of the social positions based classifier and the LDA-based approach can be measured using the evaluation metric F1 based on the gold standard mapping. The social positions' classifier achieved an F1 score of 0.77 compared with a 0.30 for the LDA-based classifier. This low F1 score is expected. The LDA-based approach only used clicked documents to estimate an LDA model. As displayed in table 6.1, the number of clicked documents per session is typically small. Also, the LDA-based classifier uses the cosine similarity score between two sessions' vectors. If the cosine score were above a pre-specified threshold, they would be considered as related. The threshold in my experiments was 0.70, which was set to only consider closely related sessions. It should be noted that the LDA-based classifier's performance can be improved by including the top 10 documents for each session regardless of whether they were clicked or not. This improved the F1 score by 100% to 0.60, which is still below the social positions' classifier by 28.33%.

6.6.5 Improvement and failure analysis

In this section, I answer the fourth research question RQ4 with regard to analysing the performance of LTR-SP and other approaches on sessions of different characteristics. The first aspect to study is session type. Starting from TREC 2012, test sessions were classified based on two facets using a framework introduced by Li and Belkin (2008). These two facets are: product and goal. The product of a search session can merely involve locating facts or information items on the web. This is called a factual product. It can also be an intellectual product when it results in new ideas or findings. The goal of the search session can be either specific or amorphous. This dimension is about the level of an information need's specificity. As suggested by Li and Belkin (2008), a specific task has a well-defined information need whereas an amorphous task will have an ill-defined need. These two facets produce four types of sessions: known-item (factual specific), interpretive (intellectual specific), known-subject (factual amorphous), and exploratory (intellectual amorphous). Across TREC 2012, 2013 and 2014 test collections, there were 112 known-item (39.30%), 56 interpretive (19.65%), 59 known-subject (20.70%), and 58 exploratory sessions (20.35%). Table 6.13 presents the performance of LTR-SP and other approaches on each of the four session's types. For each approach and evaluation metric, the percentile change compared with its mean over all sessions is reported.

LTR-SP was the best performing approach across all session types under nDCG@10 and nERR@10. It also achieved the best scores under the MAP metric for all types except for factual specific sessions. This is likely to be caused by the fact that LTR-SP uses category B of ClueWeb12 for TREC 2013 whereas the best TREC system is taking advantage of the full collection. All approaches seem to be excelling for exploratory sessions, which is expected due to the task nature. The task of session search focuses on utilising a session's data to improve performance for the current query. In exploratory search, users' information needs are ill-defined. Their task products are intellectual. Therefore, they refine and proceed in their session based on the results that are shown to them and the interaction they might have made with such results. All approaches take advantage of the interaction data and, therefore, these types of sessions seem to benefit the most. QCM, the MDP-based approach, models the session as a sequential decision making process which seems to be better suited to sessions of an exploratory nature.

For all approaches, sessions with well-defined information needs appear to benefit the least, particularly for intellectual tasks. One possible explanation is that the additional information used by such approaches may cause a drift from the actual need. The good performance on amorphous sessions comes at a risk of drifting for sessions with specific goals. This is, perhaps, one advantage of LTR-SP, which takes a risk-averse approach. Firstly, social expansion terms are extracted in a way that ensures their relevance to both the session and the session's social position to avoid including extraneous terms. Special subsets of these terms are extracted based on their appearance in clicked documents' titles or snippets. These subsets were shown to produce effective features, as shown in table 6.12. Secondly, LTR-SP uses a relatedness classifier that is more fine-grained in identifying related sessions than the LDA-based approach. Topical classification as in LTR-LDA performs closely to LTR-SP on amorphous sessions because of their exploratory nature but less so when the information need is well-defined.

Table 6.14 displays the distribution of session types per test collection. Whilst TREC

	nDCG@10	**nERR@10**	**MAP**
Factual specific (known-item)			
Initial ranker	0.2353 (0.77%)	0.2810 (-1.92%)	0.1070 (9.74%)
Best TREC	0.2550 (0.63%)	0.2798 (-6.98%)	**0.1159** (13.29%)
QCM	0.2132 (-5.37%)	0.2621 (-7.94%)	0.0786 (3.69%)
LTR-Base	0.2659 (-3.76%)	0.3332 (-4.96%)	0.1058 (4.13%)
LTR-LDA	0.2798 (-3.62%)	0.3596 (-2.99%)	0.1072 (2.68%)
LTR-SP	**0.2932** (-3.93%)	**0.3780** (-4.23%)	0.1111 (0.27%)
Factual amorphous (known-subject)			
Initial ranker	0.2081 (-10.88%)	0.2551 (-10.96%)	0.1082 (10.97%)
Best TREC	0.2720 (7.34%)	0.3437 (14.26%)	0.1233 (20.53%)
QCM	0.2215 (-1.69%)	0.2850 (0.11%)	0.0920 (21.37%)
LTR-Base	0.2839 (2.75%)	0.3822 (9.01%)	0.1285 (26.48%)
LTR-LDA	0.3119 (7.44%)	0.4036 (8.88%)	0.1336 (27.97%)
LTR-SP	**0.3203** (4.95%)	**0.4196** (6.31%)	**0.1462** (31.95%)
Intellectual specific (interpretive)			
Initial ranker	0.2011 (-13.88%)	0.2712 (-5.34%)	0.0638 (-34.56%)
Best TREC	0.2194 (-13.42%)	0.2849 (-5.29%)	0.0522 (-48.97%)
QCM	0.1791 (-20.51%)	0.2396 (-15.84%)	0.0358 (-52.77%)
LTR-Base	0.2474 (-10.46%)	0.3210 (-8.44%)	0.0629 (-38.09%)
LTR-LDA	0.2514 (-13.40%)	0.3329 (-10.20%)	0.0645 (-38.22%)
LTR-SP	**0.2811** (-7.90%)	**0.3723** (-5.68%)	**0.0716** (-35.38%)
Intellectual amorphous (exploratory)			
Initial ranker	0.2870 (22.91%)	0.3437 (19.97%)	0.1010 (3.59%)
Best TREC	0.2640 (4.18%)	0.3130 (4.06%)	0.1033 (0.98%)
QCM	0.2974 (32%)	0.3715 (30.49%)	0.0926 (22.16%)
LTR-Base	0.3163 (14.48%)	0.3805 (8.53%)	0.1035 (1.87%)
LTR-LDA	0.3265 (12.47%)	0.3951 (6.58%)	0.1078 (3.26%)
LTR-SP	**0.3362** (10.16%)	**0.4234** (7.27%)	**0.1123** (1.35%)

Table 6.13: Search accuracy on four session types. For each approach and evaluation metric, the percentile change compared with its mean over all sessions is reported.

2011 sessions were not classified, the majority of them were known-item sessions. This distribution can explain LTR-SP and LTR-LDA performance in tables 6.8 and 6.9. Both TREC 2011 and TREC 2013 were dominated by sessions with specific goals (71.26% of sessions). LTR-SP outperformed LTR-LDA by 7.75% on TREC 2011 and by 16.49% on TREC 2013. For TREC 2012 and 2014, the proportion of amorphous sessions was higher

	TREC 2012	TREC 2013	TREC 2014
Factual specific	43 (43.87%)	48 (55.17%)	21 (21%)
Factual amorphous	20 (20.41%)	12 (13.80%)	27 (27%)
Intellectual specific	12 (12.24%)	14 (16.09%)	30 (30%)
Intellectual amorphous	23 (23.47%)	13 (14.94%)	22 (22%)

Table 6.14: Distribution of search session types in TREC Session tracks of 2012, 2013 and 2014.

by, 43.88% and 49%, respectively. LTR-LDA favoured such sessions and thus is less behind LTR-SP by 2.20% on TREC 2012 and 3.47% on TREC 2014.

The second dimension to investigate was session length. In figure 6.4, I plotted the performance of a number of approaches based on session length. The test collections included 106 sessions of two queries length, 85 sessions of length 3, 55 with four queries, 45 of length 5, and 70 sessions with six or more queries. In each group, approximately 60% sessions had specific goals and 40% were amorphous. There are a number of observations to be made from this graph. Firstly, all approaches seem to degrade for longer sessions (6+ queries). This is perhaps a sign of struggle sessions where the observer search engine is not returning relevant documents. Aula et al. (2010) found that users tend to formulate more queries when they face difficulty in locating relevant information. Secondly, LTR-based approaches are, obviously, dependent on the initial ranker performance. They show a stable performance with sessions of length below 6 except for three-queries sessions. The initial ranker does not return any relevant document for 18% of sessions with length 3. As a result, there is a slight drop in performance for these approaches under this category. In comparison, the initial ranker fails to retrieve relevant documents for only 7% of sessions with length 4, which does not greatly affect the steady performance of dependent approaches. Thirdly, the QCM approach seems to improve as the sessions' lengths increase. It reaches its peak for sessions with 5 queries. However, two thirds of all test sessions are less than 5 queries in length. In fact, half of the test sessions are either 2 or 3 queries in length.

6.7 Summary

In this chapter, I presented LTR-SP, which is a novel session search approach based on social positions. Session search aims to improve search accuracy for a test query using previous user's interactions at the session level. LTR-SP consisted of four components: an initial ranker, a social position matcher, a relatedness classifier, and a learning to rank model. In section 6.4.1, I introduced a novel query formulation method for use as the initial ranker. This method interpolated a session's concatenated query with a selected relevance model. The second component, described in section 6.4.2, mapped each test session to its

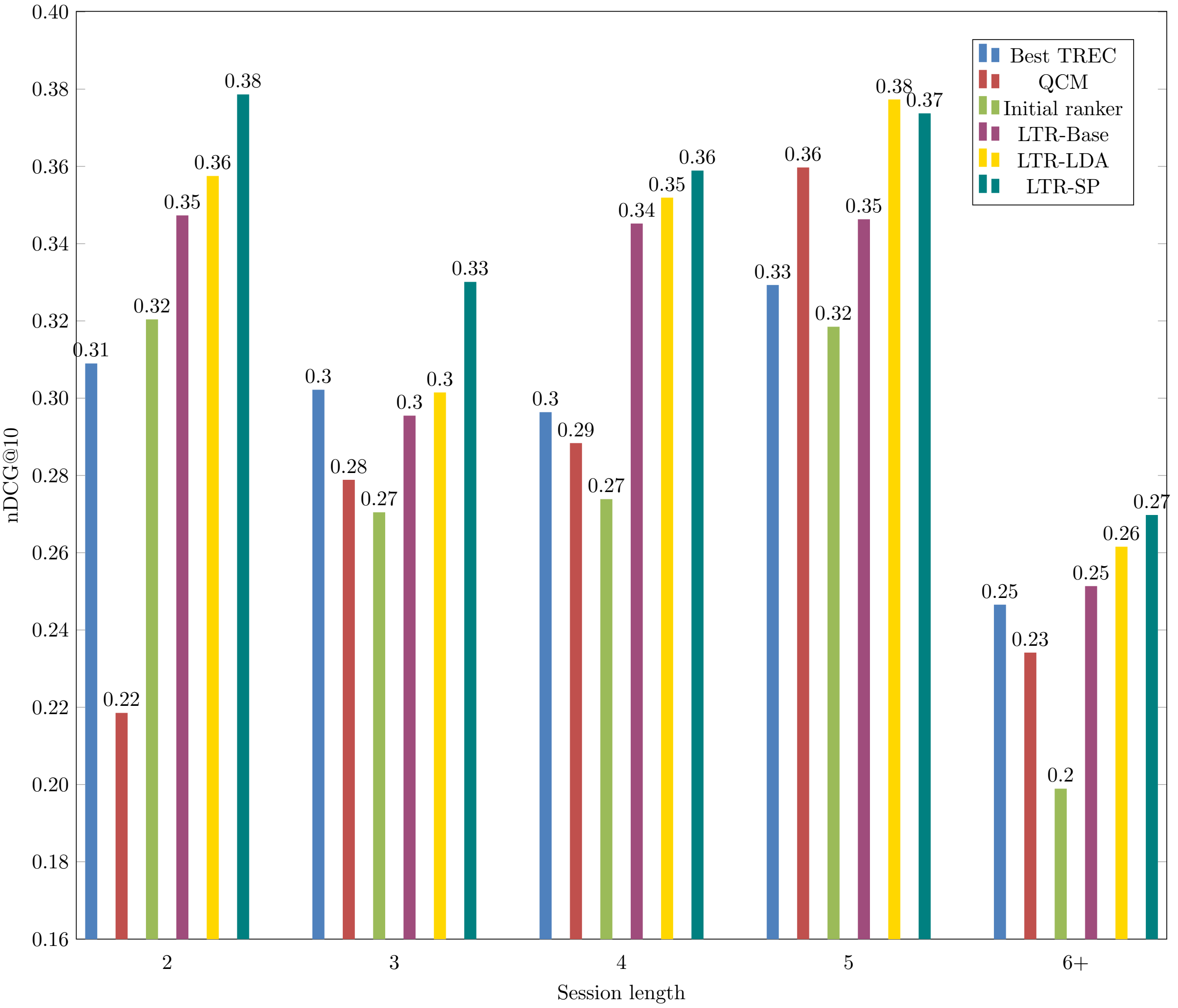


Figure 6.4: Search accuracy as measured by nDCG@10 for sessions with varying number of queries.

relevant social positions using the user models that were estimated based on methods in chapter 4. A relatedness classifier was introduced in section 6.4.3 to identify other sessions in query logs that are likely to be on the same information need. This classifier relied on identifying the social positions of each search session and used a small set of novel features. The fourth component was a learning to rank model based on lambdaMART. It used a set of social positions' features and another set of general features. LTR-SP was shown to be effective in improving performance for the task of session-bases search. A detailed investigation of LTR-SP performance was provided in section 6.6.

CHAPTER 7

Conclusions

In this thesis, I proposed a framework to model search engine users. Rather than estimating a user model for each individual user in an ad-hoc manner, I suggested a principled approach using webpages as a source to learn about possible users' interests. This approach has an integrated advantage over other personalisation methods in that it does not require real users' data. An important implication of such a method is that the privacy of users could be preserved by design. Inspired by social role theory, the suggested framework builds on the concept of social positions. A social position is a label for a community of users with similar interests. There are three main phases of the framework: social position identification, representation, and matching. I have presented novel and minimally supervised machine learning methods to extract and represent social positions. I also proposed two effective approaches to web search tasks: diversification and session-based search. The first task utilised query-based matching to determine the relevant social positions for a search query while matching in the second task was used to identify social positions for a search session. This chapter provides a summary of this thesis by first describing the framework's components in section 7.1 then reporting my key findings in section 7.2. Section 7.3 concludes the thesis by discussing potential future research directions.

7.1 Overview of framework's components

Social position identification The first phase of the proposed framework was to identify social positions which was formulated as a binary classification task. The input to the classifier was a set of candidate nouns or noun phrases. To enumerate such a set, I proposed a linguistic pattern matching method. This method relied on patterns commonly used by people to associate themselves or others to a specific social position.

Social position representation The goal of this component was to build a computational representation for each social position. This was achieved using Differential

Latent Dirichlet Allocation (DiffLDA), which is a topic modelling approach that represents each social position as a topic, i.e. a multinomial distribution over words. Each topic was estimated from a position's document collection. These collections were built using an unsupervised approach in two steps: extracting seed terms for a social position and retrieving documents from a web document collection using the social position and its seed terms as a weighted query.

Query-based matching A query-based matching component involves identifying relevant social positions for a search query. Search results diversification is a prominent application for such a component. To diversify search results, a diversification algorithm requires a representation of the multiple possible interpretations and aspects of a search query. I proposed a diversification method based on determining relevant social positions for ambiguous or underspecified queries. These social positions represent the different users who might have submitted the query. I suggested a method to estimate the relevance of candidate social positions for a search query using the query's top results and then diversification was approached as a two-step process. Firstly, a diversified list was constructed for each social position which included documents relevant to the social position. Such a list would cover multiple aspects of a search query from a viewpoint of a specific social position. Secondly, the final ranked list of results was proportionally diversified based on social positions' importance.

Session-based matching A web search session constitutes multiple interactions that are performed by a user to fulfil a single information need (Jansen et al., 2007a). It is perhaps natural to assume that such a user would belong to a single or few social positions. This component attempted to determine social positions relevant to a search session. I designed this component in the context of session-based search; a task to improve a user's search experience by utilising their previous interactions to improve performance for the user's next query at the session level. I formulated this task under the framework of learning to rank. Firstly, I proposed a method to match social positions to a search session based on a measure of semantic similarity. Secondly, I presented a classification-based approach to identify related sessions to a test session from query logs. Thirdly, I developed a set of novel features that were derived from the session's social positions and related sessions. These features, along with general features, were used to train lambdaMART, as the learning to rank algorithm.

7.2 Key findings

On the identification of social position I presented a linguistic pattern matching approach to enumerate a set of candidate positions. This method resulted in the extraction of $204,781$ candidates. A sample was drawn from this set and labelled by four annotators to identify social positions. A reasonable inter-annotator agreement was observed ($k = 0.77$), suggesting that users would often agree on identifying social positions. I, therefore, developed a binary classification model based on the AROW algorithm to identify social positions. The proposed classifier achieved an accuracy of 85.8% using a set of novel features that were collected based on parsed web sentences. This result indicates that this task can be automated.

On the representation of social position I proposed to represent a social position as a topic using DiffLDA, a novel topic modelling approach. DiffLDA was compared with relevance modelling as a baseline. The goal of these two approaches was to construct *a coherent* set of words that could be used as a representation of a social position. DiffLDA demonstrated significantly improved performance relative to relevance modelling under two coherence measures: NPMI and PMI. When investigating the performance of DiffLDA, a strong correlation between its coherence and the coherence of seed terms was observed. This suggested that a coherent topic could be inferred for a social position given a small set of coherent seed terms.

On search results diversification I designed a diversification approach that used social positions to represent a search query's possible interpretations. My experiments on the diversity task of TREC web track demonstrated the effectiveness of social position compared with other implicit diversification approaches. The proposed approach was the only method to provide a statistically significant improvement over the query likelihood (QL) baseline. The relative improvement compared with QL were +12.5%, +8%, and +15% under ERR-IA, α-NDCG and NRBP, respectively. Robustness analysis showed that all considered approaches helped test queries with a comparable rate. However, the proposed approach had the lowest impact on hurt queries. The proposed approach also offered several performance advantages over other implicit approaches. Firstly, all considered baselines required the number of possible clusters, subtopics, behind a search query to be set a priori while the proposed approach estimated such a number based on a query-based matching method. Secondly, documents that were not related to a relevant social position were not considered in the final ranked list. This removal process was shown to be useful in eliminating non-relevant documents. Further analysis demonstrated that the proposed approach provided significant improvement relative to the QL baseline for both faceted and ambiguous queries while other approaches seemed to only provide marginal improvement for faceted queries.

On session-based search I proposed LTR-SP, which is a learning to rank model based on social positions' features designed for the task of session-based search. LTR-SP outperformed a number of state-of-the-art session search systems on a number of evaluation metrics. I obtained an average improvement relative to the best TREC systems of 17% and 24% on nDCG@10 and nERR@10, respectively. Unlike other approaches, the model has also shown consistent performance over four test collections from the TREC session track. Analysis suggested that the model's strong performance is mainly attributed to learning features derived from a session's social positions. Furthermore, features that were extracted from related sessions have shown to be useful in improving performance. LTR-SP performance was also stable across various types of sessions and was favourable when compared with other approaches. Session-based search systems, including LTR-SP, performed better on sessions with ill-defined information needs.

7.3 Future work

In this thesis, I developed approaches to match social positions to a search query or a session. It would be equally important to study the matching of social positions to real users based on a user's browsing activity. In such a case, a user model would consist of a distribution over social positions relevant to the user's visited webpages. The proposed framework could potentially offer several benefits. Firstly, it is transparent in the sense that users could easily grasp the semantics of social positions and the reasons based on which certain webpages are ranked higher than others for them. Secondly, users can scrutinise their model and update it by simply adding or removing social positions that best match their interests. Thirdly, the framework learns user models for each social position from public webpages and independently from any real user's interaction. This could be potentially useful in building a client-side personalisation agent that stores a user's membership data related to social positions locally in the user's device.

Additionally, previous research on social role theory and social networks analysis assume that people can easily recognise social positions and associate them with their respective social roles or characteristic behaviours (Biddle, 2013; Wasserman and Faust, 2009). Such a hypothesis could be tested via a user study to assess the interpretability of the proposed user models. The interpretability of such models could be analysed from three different perspectives: the social position, the social role, and the learning algorithm. The first assesses a user's understanding of a social position as a label for a set of related information activities or behaviours. In theory, it should be easy for a user to immediately recognise a noun or a noun phrase as a social position. However, my study on social position identification showed that annotators were not in total agreement about some candidate social positions. The second aspect of interpretability relates to the computational representation of a social position, i.e. the social role. In my thesis, I represented a social position as a multinomial distribution over words. I evaluated such representations based on their coherence but not on their relevance to social positions or their users' interpretability. Previous research on topic modelling evaluation suggests that users can easily spot incoherent representations or *intruder* terms in a set of coherent terms (Chang et al., 2009). However, it would be essential to assess the relevance of learned representations to social positions and not only its coherence.

The third perspective of interpretability is concerned with the learning model. In this thesis, I proposed a topic modelling based approach to learn social roles. These models are statistically explainable to knowledgeable audience and may even be simplified to the public. However, careful consideration needs to be taken when using other representation learning approaches such as those based on deep neural networks.

In this thesis, I studied textual IR with a focus on English text. Some of the approaches that I developed are language-specific, e.g. social position identification. As mentioned previously, one of the main propositions of role theory stipulates that roles exist in societies because of their functions and their integrated role in a social system (Biddle, 2013). Thus, it is perhaps safe to assume that social positions are also expressed in different languages and representations of such positions could be learned from sources in different languages. A cross-lingual representation of social position would be an interesting research topic for two reasons. Firstly, while social positions are perhaps shared among different societies, their characteristic behaviours might not. From the computational social science perspective, it would be valuable to examine how different social positions are depicted in different languages. Secondly, a cross-lingual representation might be a venue to combat biases and ethically inappropriate representations for social positions. This might be achieved by reducing dependence on a single source language and its models or tools.

Another area of future research is the representation component of the framework. Two aspects of this component could be further improved: the learning model and the representation structure. A social position could be represented as an embedding rather than a topic. Such a representation could be built via at least two approaches. Embeddings of each word in a social position's model could be learned via neural word embedding models. These embeddings can then be merged to form a single representation of a social position. Alternatively, since each social position is a noun or a noun phrase, it would also be possible to learn embeddings directly for social positions. An embedding representation would be simpler to integrate into the various components of my proposed framework. For example, in matching a search session, query, or document to a social position. However, embeddings might not be interpretable to users.

In terms of representation structure, I presented methods to learn flat representations of both social positions and their topics. Social positions can be represented in a hierarchical structure to express relatedness and specificity of one social position to another. For example, social positions such as *computer science student, physics student and medical student* are expected to share some topical interests since they are all *students*. A number of topics, which could also have a hierarchical representation, can also be induced for each social position instead of one. Finally, a social position could be represented by search tasks and not just topics of interests. One major advantage of the proposed approach is its use of web documents as a resource to learn about users' behaviour. By labelling some webpages as relevant to certain social position, they can be further processed using advanced natural language processing methods to identify and extract search tasks that are relevant to the social position.

Bibliography

Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26.

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining*, pages 5–14.

Ahn, J., Brusilovsky, P., He, D., Grady, J., and Li, Q. (2008). Personalized web exploration with task models. In *Proceedings of the 17th international conference on World Wide Web*, pages 1–10.

Aktolga, E. (2014). *Integrating non-topical aspects into information retrieval*. PhD thesis, University of Massachusetts Amherst.

Aktolga, E. and Allan, J. (2013). Sentiment diversification with different biases. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 593–602.

Albakour, M., Kruschwitz, U., Niu, J., and Fasli, M. (2010). University of essex at the TREC 2010 session track. In *Proceedings of The Nineteenth Text REtrieval Conference, TREC*.

Aletras, N. and Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22.

Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D. J., et al. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. In *ACM SIGIR Forum*, volume 37, pages 31–47. ACM New York, NY, USA.

Allan, J., Croft, W. B., Moffat, A., and Sanderson, M. (2012). Frontiers, challenges, and opportunities for information retrieval: Report from SWIRL 2012 the second strategic workshop on information retrieval in lorne. In *SIGIR Forum*, volume 46, pages 2–32.

Allen, B. (1996). *Information Tasks: Toward a User-Centered Approach to Information Systems*. Academic Press, Inc.

Allen, D., Karanasios, S., and Slavova, M. (2011). Working with activity theory: Context, technology, and information behavior. *Journal of the American Society for Information Science and Technology*, 62(4):776–788.

Aloteibi, S. and Sanderson, M. (2014). Analyzing geographic query reformulation: An exploratory study. *Journal of the association for information science and technology*, 65(1):13–24.

Anick, P. (2003). Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 88–95.

Ashkan, A. and Clarke, C. L. (2011). On the informativeness of cascade and intent-aware effectiveness measures. In *Proceedings of the 20th international conference on World wide web*, pages 407–416.

Aula, A. (2003). Query formulation in web information search. In *Proceedings of the IADIS International Conference WWW/Internet 2003*, pages 403–410.

Aula, A., Khan, R. M., and Guan, Z. (2010). How does search behavior change as search becomes more difficult? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 35–44.

Aula, A. and Nordhausen, K. (2006). Modeling successful performance in web searching. *Journal of the american society for information science and technology*, 57(12):1678–1693.

Azzopardi, L. (2005). *Incorporating context within the language modeling approach for ad hoc information retrieval*. PhD thesis, University of Paisley.

Azzopardi, L. (2009). Query side evaluation: an empirical analysis of effectiveness and effort. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 556–563.

Azzopardi, L. (2011). The economics in interactive information retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 15–24.

Azzopardi, L., Kelly, D., and Brennan, K. (2013). How query cost affects search behavior. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 23–32.

Baeza-Yates, R. and Ribeiro-Neto, B. A. (2011). *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd.

Baeza-Yates, R. A., Hurtado, C. A., and Mendoza, M. (2004). Query recommendation using query logs in search engines. In *Current Trends in Database Technology - EDBT 2004 Workshops*, pages 588–596. Springer.

Bamman, D. (2015). *People-Centric Natural Language Processing.* PhD thesis, Carnegie Mellon University.

Baskaya, F., Keskustalo, H., and Järvelin, K. (2013). Modeling behavioral factors ininteractive information retrieval. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2297–2302.

Beeferman, D. and Berger, A. (2000). Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 407–416.

Belkin, N. J., Cole, M. J., Gwizdka, J., Li, Y., Liu, J., Muresan, G., Smith, C. A., Taylor, A., Yuan, X., and Roussinov, D. (2005). Rutgers information interaction lab at TREC 2005: Trying HARD. In *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005*, volume 500-266.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Bendersky, M. and Croft, W. B. (2008). Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 491–498.

Bendersky, M., Croft, W. B., and Diao, Y. (2011a). Quality-biased ranking of web documents. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 95–104.

Bendersky, M., Croft, W. B., and Smith, D. A. (2011b). Joint annotation of search queries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 102–111.

Bendersky, M., Metzler, D., and Croft, W. B. (2010). Learning concept importance using a weighted dependence model. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 31–40.

Bendersky, M., Metzler, D., and Croft, W. B. (2011c). Parameterized concept weighting in verbose queries. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 605–614.

Bennett, P. N., Radlinski, F., White, R. W., and Yilmaz, E. (2011). Inferring and using location metadata to personalize web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 135–144.

Bennett, P. N., White, R. W., Chu, W., Dumais, S. T., Bailey, P., Borisyuk, F., and Cui, X. (2012). 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.

Bi, B., Shokouhi, M., Kosinski, M., and Graepel, T. (2013). Inferring the demographics of search users: Social data meets search queries. In *Proceedings of the 22nd international conference on World Wide Web*, pages 131–140.

Biddle, B. J. (1986). Recent developments in role theory. *Annual review of sociology*, 12(1):67–92.

Biddle, B. J. (2013). *Role theory: Expectations, identities, and behaviors.* Academic Press.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Boldi, P., Bonchi, F., Castillo, C., and Vigna, S. (2009). From "dango" to "japanese cakes": Query reformulation models and patterns. In *2009 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 183–190. IEEE Computer Society.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of GSCL*, pages 31–40.

Boyd-Graber, J., Hu, Y., Mimno, D., et al. (2017). Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296.

Brajnik, G., Guida, G., and Tasso, C. (1990). User modeling in expert man-machine interfaces: A case study in intelligent information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(1):166–185.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

Broder, A. (2002). A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM New York, NY, USA.

Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. (2000). Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659.

Brusilovsky, P. (2001). Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2):87–110.

Brusilovsky, P. and Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 3–53. Springer.

Bruza, P. and Dennis, S. (1997). Query reformulation on the internet: Empirical data and the hyperindex search engine. In Devroye, L. and Chrisment, C., editors, *Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 1997, 5th International Conference*, pages 488–500.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. Technical report, Microsoft Research.

Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2010). *Information Retrieval - Implementing and Evaluating Search Engines*. MIT Press.

Cai, F., Liang, S., and De Rijke, M. (2014). Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 835–838.

Callan, J. (2012). The lemur project and its clueweb12 dataset. In *Invited talk at the SIGIR 2012 Workshop on Open-Source Information Retrieval*.

Callan, J., Allan, J., Clarke, C. L., Dumais, S., Evans, D. A., Sanderson, M., and Zhai, C. (2007). Meeting of the minds: An information retrieval research agenda. In *ACM SIGIR Forum*, volume 41, pages 25–34. ACM New York, NY, USA.

Callan, J., Hoy, M., Yoo, C., and Zhao, L. (2009). The clueweb09 dataset. `http://lemurproject.org/clueweb09/index.php`. Online; accessed 9 December 2020.

Carbonell, J. G. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. ACM.

Carman, M. J., Crestani, F., Harvey, M., and Baillie, M. (2010). Towards query log based personalization using topic models. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1849–1852.

Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'El, N., Ronen, I., Uziel, E., Yogev, S., and Chernov, S. (2009). Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1227–1236.

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50.

Carroll, J. M. and Rosson, M. B. (1987). Paradox of the active user. In *Interfacing thought: Cognitive aspects of human-computer interaction*, pages 80–111.

Carterette, B., Bah, A., Kanoulas, E., Hall, M. M., and Clough, P. D. (2013). Overview of the TREC 2013 session track. In *Proceedings of The Twenty-Second Text REtrieval Conference, TREC*.

Carterette, B. and Chandar, P. (2009). Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, page 12871296.

Carterette, B., Clough, P., Hall, M., Kanoulas, E., and Sanderson, M. (2016). Evaluating retrieval over sessions: The trec session track 2011-2014. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 685–688.

Carterette, B., Kanoulas, E., Hall, M. M., and Clough, P. D. (2014). Overview of the TREC 2014 session track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC*.

Carterette, B., Pavlu, V., Fang, H., and Kanoulas, E. (2009). Million query track 2009 overview. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC.*

Case, D. (2002). Looking for information: a survey of research on information seeking, needs, and behavior. *Library and information science.*

Chambers, N. W. (2011). *Inducing Event Schemas and their Participants from Unlabeled Text.* PhD thesis, Stanford University.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22:288–296.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630.

Chapelle, O. and Zhang, Y. (2009). A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10.

Chemudugunta, C., Smyth, P., and Steyvers, M. (2006). Modeling general and specific aspects of documents with a probabilistic topic model. *Advances in neural information processing systems*, 19:241–248.

Chen, D., Chen, W., Wang, H., Chen, Z., and Yang, Q. (2012a). Beyond ten blue links: enabling user click modeling in federated web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 463–472.

Chen, H. and Dumais, S. (2000). Bringing order to the web: Automatically categorizing search results. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 145–152.

Chen, H. and Karger, D. R. (2006). Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 429–436.

Chen, J., Liu, Y., Luo, C., Mao, J., Zhang, M., and Ma, S. (2018). Improving session search performance with a multi-mdp model. In *Asia Information Retrieval Symposium*, pages 45–59. Springer.

Chen, Z., Wei, M., Nan, J., Chen, J., Yu, X., Liu, Y., and Cheng, X. (2012b). ICTNET at session track TREC 2012. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC.*

Chin, D. N. (1989). Knome: Modeling what the user knows in uc. In *User models in dialog systems*, pages 74–107. Springer.

Chirita, P.-A., Firan, C. S., and Nejdl, W. (2007). Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14.

Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using odp metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185.

Choueka, Y. and Lusignan, S. (1985). Disambiguation by short contexts. *Computers and the Humanities*, 19(3):147–157.

Chuklin, A., Markov, I., and Rijke, M. d. (2015). Click models for web search. *Synthesis lectures on information concepts, retrieval, and services*, 7(3):1–115.

Clark, S. (2015). Vector space models of lexical meaning. *The Handbook of Contemporary semantic theory*, pages 493–522.

Clark, S. and Curran, J. R. (2007). Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33(4):493–552.

Clark, S. and Weir, D. (2002). Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Clarke, C. L., Agichtein, E., Dumais, S., and White, R. W. (2007). The influence of caption features on clickthrough patterns in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 135–142.

Clarke, C. L., Craswell, N., Soboroff, I., and Ashkan, A. (2011a). A comparative analysis of cascade measures for novelty and diversity. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 75–84.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666.

Clarke, C. L., Kolla, M., and Vechtomova, O. (2009a). An effectiveness measure for ambiguous and underspecified queries. In *Conference on the Theory of Information Retrieval*, pages 188–199. Springer.

Clarke, C. L. A., Craswell, N., and Soboroff, I. (2009b). Overview of the TREC 2009 web track. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009*.

Clarke, C. L. A., Craswell, N., Soboroff, I., and Cormack, G. V. (2010). Overview of the TREC 2010 web track. In *Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010*.

Clarke, C. L. A., Craswell, N., Soboroff, I., and Voorhees, E. M. (2011b). Overview of the TREC 2011 web track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011*.

Clarke, C. L. A., Craswell, N., and Voorhees, E. M. (2012). Overview of the TREC 2012 web track. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC*.

Clauset, A., Shalizi, C. R., and Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4):661–703.

Cleverdon, C. W. (1959). The evaluation of systems used in information retrieval. In *Proceedings of the international conference on scientific information*, volume 1, pages 687–698. National Academy of Sciences Washington, DC.

Clough, P., Sanderson, M., Abouammoh, M., Navarro, S., and Paramita, M. (2009). Multiple approaches to analysing query diversity. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 734–735.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Cormack, G. V., Smucker, M. D., and Clarke, C. L. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465.

Courtright, C. (2007). Context in information behavior research. *Annual review of information science and technology*, 41(1):273–306.

Crammer, K., Kulesza, A., and Dredze, M. (2009). Adaptive regularization of weight vectors. In *Advances in neural information processing systems*, pages 414–422.

Crammer, K. and Singer, Y. (2002). Pranking with ranking. In *Advances in neural information processing systems*, pages 641–647.

Craswell, N., Zoeter, O., Taylor, M. J., and Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 87–94. ACM.

Croft, W. B., Metzler, D., and Strohman, T. (2010). *Search engines: Information retrieval in practice*. Addison-Wesley.

Cucerzan, S. and White, R. W. (2007). Query suggestion based on user landing pages. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 875–876.

Dai, Z., Xiong, C., Callan, J., and Liu, Z. (2018). Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134.

Dang, H. T., Kelly, D., and Lin, J. J. (2007). Overview of the TREC 2007 question answering track. In *Proceedings of The Sixteenth Text REtrieval Conference, TREC*.

Dang, V. and Croft, B. W. (2010). Query reformulation using anchor text. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 41–50.

Dang, V. and Croft, B. W. (2013). Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 603–612.

Dang, V. and Croft, W. B. (2012). Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 65–74.

Das, A. S., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dehghani, M., Rothe, S., Alfonseca, E., and Fleury, P. (2017). Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1747–1756.

Donato, D., Bonchi, F., Chi, T., and Maarek, Y. (2010). 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.

Dou, Z., Song, R., and Wen, J.-R. (2007). A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590.

Downey, D., Dumais, S., Liebling, D., and Horvitz, E. (2008). Understanding the relationship between searchers' queries and information goals. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 449–458.

Dredze, M., Crammer, K., and Pereira, F. (2008). Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, pages 264–271.

Dupret, G. E. and Piwowarski, B. (2008). A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338.

El-Arini, K., Paquet, U., Herbrich, R., Van Gael, J., and Agüera y Arcas, B. (2012). Transparent user models for personalization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 678–686.

Feng, Y., Xu, J., Lan, Y., Guo, J., Zeng, W., and Cheng, X. (2018). From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 125–134.

Finin, T. W. (1989). Gumsa general user modeling shell. In *User models in dialog systems*, pages 411–430. Springer.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Fleiss, J. L., Levin, B., and Paik, M. C. (2003). *Statistical methods for rates and proportions*. John Wiley & Sons, Inc.

Fox, S., Karnawat, K., Mydland, M., Dumais, S., and White, T. (2005). Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969.

Freyne, J. and Smyth, B. (2006). Cooperating search communities. In *Adaptive Hypermedia and Adaptive Web-Based Systems, 4th International Conference, AH 2006*, pages 101–110. Springer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Fuxman, A., Tsaparas, P., Achan, K., and Agrawal, R. (2008). Using the wisdom of the crowds for keyword generation. In *Proceedings of the 17th international conference on World Wide Web*, pages 61–70.

Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, pages 54–89. Springer.

Ge, S., Dou, Z., Jiang, Z., Nie, J.-Y., and Wen, J.-R. (2018). Personalizing search results using hierarchical rnn with query-aware attention. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 347–356.

Ghorab, M. R., Zhou, D., Oconnor, A., and Wade, V. (2013). Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443.

Gillespie, C. S. (2014). Fitting heavy tailed distributions: the powerlaw package. *arXiv preprint arXiv:1407.3492*.

Gollapudi, S. and Sharma, A. (2009). An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390.

Griffiths, T., Jordan, M., Tenenbaum, J., and Blei, D. (2003). Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17–24.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.

Guan, D. and Yang, H. (2014). Is the first query the most important: An evaluation of query aggregation schemes in session search. In *Asia Information Retrieval Symposium*, pages 86–99. Springer.

Guan, D., Yang, H., and Goharian, N. (2012). Effective structured query formulation for session search. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC*.

Guan, D., Zhang, S., and Yang, H. (2013). Utilizing query change for session search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 453–462.

Guo, F., Liu, C., and Wang, Y. M. (2009). Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*, pages 124–131.

Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64.

Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wu, C., Croft, W. B., and Cheng, X. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.

Guo, J., Xu, G., Li, H., and Cheng, X. (2008). A unified and discriminative model for query refinement. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008*, pages 379–386. ACM.

Guo, Q. and Agichtein, E. (2008). 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.

Hafernik, C. T. and Jansen, B. J. (2013). Understanding the specificity of web search queries. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 1827–1832. ACM.

Hanani, U., Shapira, B., and Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User modeling and user-adapted interaction*, 11(3):203–259.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Harvey, M., Crestani, F., and Carman, M. J. (2013). Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2309–2314.

Hassan, A., White, R. W., Dumais, S. T., and Wang, Y.-M. (2014). Struggling or exploring? disambiguating long search sessions. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 53–62.

He, J., Hollink, V., and de Vries, A. (2012). Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 851–860.

He, J., Meij, E., and de Rijke, M. (2011). Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*, 62(3):550–571.

Hearst, M. (2009). *Search user interfaces.* Cambridge university press.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *International Conference on Theory and Practice of Digital Libraries*, pages 569–584. Springer.

Hoeber, O. and Yang, X. D. (2006). The visual exploration ofweb search results using hotmap. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 157–165. IEEE.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Hölscher, C. and Strube, G. (2000). Web search behavior of internet experts and newbies. *Computer networks*, 33(1-6):337–346.

Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Hu, J., Zeng, H.-J., Li, H., Niu, C., and Chen, Z. (2007). Demographic prediction based on user's browsing behavior. In *Proceedings of the 16th international conference on World Wide Web*, pages 151–160.

Huang, J. and Efthimiadis, E. N. (2009). Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 77–86.

Huang, J., White, R. W., Buscher, G., and Wang, K. (2012). 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.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Hui, K., Yates, A., Berberich, K., and de Melo, G. (2017). Pacrr: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1058.

Ieong, S., Mishra, N., Sadikov, E., and Zhang, L. (2012). Domain bias in web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 413–422.

Ingwersen, P. and Järvelin, K. (2005). *The turn: Integration of information seeking and retrieval in context.* Springer.

Jansen, B. J., Booth, D. L., and Spink, A. (2009). Patterns of query reformulation during web searching. *Journal of the american society for information science and technology*, 60(7):1358–1371.

Jansen, B. J., Spink, A., Blakely, C., and Koshman, S. (2007a). Defining a session on web search engines. *The Journal of the Association for Information Science and Technology*, 58(6):862–871.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227.

Jansen, B. J., Zhang, M., and Spink, A. (2007b). Patterns and transitions of query reformulation during web searching. *Int. J. Web Inf. Syst.*, 3(4):328–340.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Järvelin, K., Price, S. L., Delcambre, L. M. L., and Nielsen, M. L. (2008). Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR*, pages 4–15. Springer.

Jiang, J. and Allan, J. (2014). Umass at trec 2014 session track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC.*

Jiang, J. and He, D. (2013). Pitt at TREC 2013: Different effects of click-through and past queries on whole-session search performance. In *Proceedings of The Twenty-Second Text REtrieval Conference, TREC.*

Jiang, J., He, D., and Han, S. (2012). On duplicate results in a search session. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC.*

Jiang, J. and Ni, C. (2016). What affects word changes in query reformulation during a task-based search session? In *Proceedings of the 2016 ACM on conference on human information interaction and retrieval*, pages 111–120.

Jiang, Z., Wen, J.-R., Dou, Z., Zhao, W. X., Nie, J.-Y., and Yue, M. (2017). Learning to diversify search results via subtopic attention. In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 545–554.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Joachims, T., Granka, L. A., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161. ACM.

Jones, R., Kumar, R., Pang, B., and Tomkins, A. (2007). " i know what you did last summer" query logs and user privacy. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914.

Kacimi, M. and Gamper, J. (2011). Diversifying search results of controversial queries. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 93–98.

Kanoulas, E., Carterette, B., Clough, P. D., and Sanderson, M. (2011a). Evaluating multi-query sessions. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1053–1062.

Kanoulas, E., Carterette, B., Hall, M. M., Clough, P. D., and Sanderson, M. (2012). Overview of the TREC 2012 session track. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC*.

Kanoulas, E., Clough, P. D., Carterette, B., and Sanderson, M. (2010). Overview of the TREC 2010 session track. In *Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010*.

Kanoulas, E., Hall, M. M., Clough, P. D., Carterette, B., and Sanderson, M. (2011b). Overview of the TREC 2011 session track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC*.

Kato, M. P., Sakai, T., and Tanaka, K. (2013). When do people use query suggestion? a query suggestion log analysis. *Information retrieval*, 16(6):725–746.

Keikha, M., Crestani, F., and Croft, W. B. (2012). Diversity in blog feed retrieval. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 525–534.

Kelly, D. (2009). Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(12):1–224.

Kelly, D., Dollu, V. D., and Fu, X. (2005). 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.

Keskustalo, H., Järvelin, K., Pirkola, A., Sharma, T., and Lykke, M. (2009). Test collection-based ir evaluation needs extension toward sessions–a case of extremely short queries. In *Asia Information Retrieval Symposium*, pages 63–74. Springer.

Kliman-Silver, C., Hannak, A., Lazer, D., Wilson, C., and Mislove, A. (2015). Location, location, location: The impact of geolocation on web search personalization. In *Proceedings of the 2015 internet measurement conference*, pages 121–127.

Kotov, A., Bennett, P. N., White, R. W., Dumais, S. T., and Teevan, J. (2011). 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.

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Sage publications.

Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–202.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119.

Lafferty, J. and Zhai, C. (2003). Probabilistic relevance models based on document and query generation. In Croft, B. and Lafferty, J., editors, *Language modeling for information retrieval*, pages 1–10. Springer.

Lau, J. H. and Baldwin, T. (2016). The sensitivity of topic coherence evaluation to topic cardinality. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–487.

Lau, J. H., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539. The Association for Computer Linguistics.

Lau, T. and Horvitz, E. (1999). Patterns of search: analyzing and modeling web query refinement. In *UM99 user modeling*, pages 119–128. Springer.

Lavrenko, V. and Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127.

Lawrie, D. J. and Croft, W. B. (2003). 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.

Lee, J. H. (1997). Analyses of multiple evidence combination. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–276.

Lee, W. M. and Sanderson, M. (2010). Analyzing url queries. *Journal of the American Society for Information Science and Technology*, 61(11):2300–2310.

Levine, N., Roitman, H., and Cohen, D. (2017). An extended relevance model for session search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 865–868.

Li, J., Song, D., Zhang, P., and Hou, Y. (2015). How different features contribute to the session search? In *Natural Language Processing and Chinese Computing*, pages 242–253. Springer.

Li, J., Zhao, X., Zhang, P., and Song, D. (2018). Modeling multiple interactions with a markov random field in query expansion for session search. *Computational Intelligence*, 34(1):345–362.

Li, P., Wu, Q., and Burges, C. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems*, 20:897–904.

Li, X., Guo, C., Chu, W., Wang, Y.-Y., and Shavlik, J. (2014). Deep learning powered in-session contextual ranking using clickthrough data. In *In Proc. of NIPS*.

Li, Y. and Belkin, N. J. (2008). A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management*, 44(6):1822–1837.

Liang, S., Ren, Z., and De Rijke, M. (2014a). Fusion helps diversification. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 303–312.

Liang, S., Ren, Z., and De Rijke, M. (2014b). Personalized search result diversification via structured learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 751–760.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384.

Linton, R. (1936). *The study of man: an introduction.* Appleton-Century.

Liu, C., Belkin, N. J., and Cole, M. J. (2012). Personalization of search results using interaction behaviors in search sessions. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 205–214.

Liu, F., Yu, C., and Meng, W. (2004). Personalized web search for improving retrieval effectiveness. *IEEE Transactions on knowledge and data engineering*, 16(1):28–40.

Liu, J., Liu, C., and Belkin, N. J. (2020). Personalization in text information retrieval: A survey. *Journal of the Association for Information Science and Technology*, 71(3):349–369.

Liu, T. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Liu, X. and Croft, W. B. (2005). Statistical language modeling for information retrieval. *Annu. Rev. Inf. Sci. Technol.*, 39(1):1–31.

Liu, Y., Song, R., Zhang, M., Dou, Z., Yamamoto, T., Kato, M. P., Ohshima, H., and Zhou, K. (2014). Overview of the ntcir-11 imine task. In *NTCIR*.

Lu, S., Dou, Z., Jun, X., Nie, J.-Y., and Wen, J.-R. (2019). Psgan: A minimax game for personalized search with limited and noisy click data. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 555–564.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

Lungely, D., Albakour, M.-D., and Kruschwitz, U. (2011). The use of domain modeling to improve performance over a query session. In *Proceedings of the ECIR*, volume 11.

Luo, J., Dong, X., and Yang, H. (2014a). Modeling rich interactions in session search - georgetown university at TREC 2014 session track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC*.

Luo, J., Dong, X., and Yang, H. (2015a). Session search by direct policy learning. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, pages 261–270.

Luo, J., Zhang, S., Dong, X., and Yang, H. (2015b). Designing states, actions, and rewards for using pomdp in session search. In *European Conference on Information Retrieval*, pages 526–537. Springer.

Luo, J., Zhang, S., and Yang, H. (2014b). Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 587–596.

Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., and Cohen, M. D. (1987). Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402.

Manning, C. D., Schütze, H., and Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge university press.

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

Matthijs, N. and Radlinski, F. (2011). 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.

Maxwell, D., Azzopardi, L., Järvelin, K., and Keskustalo, H. (2015). Searching and stopping: An analysis of stopping rules and strategies. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 313–322.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Mei, Q. and Church, K. (2008). Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 45–54.

Metzler, D. and Croft, W. B. (2005). A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479.

Micarelli, A. and Sciarrone, F. (2004). Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR*.

Miller, D. R., Leek, T., and Schwartz, R. M. (1999). A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221.

Miller, G. A. (1951). *Language and communication.* McGraw-Hill.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mimno, D., Wallach, H., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272.

Mitra, B., Craswell, N., et al. (2018). An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126.

Mitra, B., Diaz, F., and Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299.

Moffat, A. and Zobel, J. (2008). Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):1–27.

Mooers, C. N. (1950). The theory of digital handling of non-numerical information and its implications to machine economics. In *Proc. Assoc. Comput. Mach. Conf.*

Nallapati, R. (2006). *The smoothed dirichlet distribution: Understanding cross-entropy ranking in information retrieval.* PhD thesis, University of Massachusetts at Amherst.

Navarro-Prieto, R., Scaife, M., and Rogers, Y. (1999). Cognitive strategies in web searching. In *Proceedings of the 5th Conference on Human Factors & the Web*, pages 43–56.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Navigli, R. and Crisafulli, G. (2010). Inducing word senses to improve web search result clustering. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 116–126.

Newman, D., Bonilla, E. V., and Buntine, W. (2011). Improving topic coherence with regularized topic models. *Advances in neural information processing systems*, 24:496–504.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). 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.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14:849–856.

Odijk, D., White, R. W., Hassan Awadallah, A., and Dumais, S. T. (2015). Struggling and success in web search. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1551–1560.

Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., and Johnson, D. (2005). Terrier information retrieval platform. In *European Conference on Information Retrieval*, pages 517–519. Springer.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., and Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 271–278. ACL.

Pass, G., Chowdhury, A., and Torgeson, C. (2006). A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*, pages 1–7.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The adaptive web*, volume 4321, pages 325–341. Springer.

Peng, J., Macdonald, C., He, B., Plachouras, V., and Ounis, I. (2007). Incorporating term dependency in the dfr framework. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 843–844.

Phan, N., Bailey, P., and Wilkinson, R. (2007). Understanding the relationship of information need specificity to search query length. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 709–710.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Psarras, I. and Jose, J. (2006). A system for adaptive information retrieval. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 313–317. Springer.

Purcell, K., Brenner, J., and Rainie, L. (2012). Search engine use 2012. Technical report, Pew Research Center.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Radlinski, F. and Dumais, S. (2006). Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692.

Radlinski, F., Kleinberg, R., and Joachims, T. (2008a). Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791.

Radlinski, F., Kurup, M., and Joachims, T. (2008b). How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52.

Radlinski, F., Szummer, M., and Craswell, N. (2010). Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on World wide web*, pages 1171–1172.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256. Association for Computational Linguistics.

Raman, K., Bennett, P. N., and Collins-Thompson, K. (2013). Toward whole-session relevance: exploring intrinsic diversity in web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 463–472.

Resnik, P. S. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania.

Rich, E. (1979). User modeling via stereotypes. *Cognitive science*, 3(4):329–354.

Rieh, S. Y. and Xie, H. I. (2006). Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management*, 42(3):751–768.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Robertson, S. E. (1977). The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304.

Robertson, S. E. and Spärck-Jones, K. (1994). Simple, proven approaches to text retrieval. Technical report, University of Cambridge, Computer Laboratory.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241. Springer.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. (1994). Okapi at TREC-3. In Harman, D. K., editor, *Proceedings of The Third Text REtrieval Conference, TREC*.

Rocchio, J. (1971). Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., and Smyth, P. (2004). The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494.

Sakai, T. (2006). Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 525–532.

Sakai, T. and Song, R. (2011). Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1043–1052.

Salton, G. (1968). *Automatic information organization and retrieval.* McGraw-Hill.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sanderson, M. (1996). *Word sense disambiguation and information retrieval.* PhD thesis, University of Glasgow.

Sanderson, M. (2008). Ambiguous queries: test collections need more sense. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 499–506.

Sanderson, M. (2010). Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375.

Sanderson, M. and Croft, W. B. (2012). The history of information retrieval research. *Proc. IEEE*, 100(Centennial-Issue):1444–1451.

Sanderson, M., Paramita, M. L., Clough, P., and Kanoulas, E. (2010). Do user preferences and evaluation measures line up? In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 555–562.

Santos, R. L., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890.

Saracevic, T. (2007). Relevance: A review of the literature and a framework for thinking on the notion in information science. part iii: Behavior and effects of relevance. *Journal of the American Society for information Science and Technology*, 58(13):2126–2144.

Schick, T., Udupa, S., and Schütze, H. (2021). Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *arXiv preprint arXiv:2103.00453*.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E. P., and Ungar, L. H. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLOS ONE*, 8(9):1–16.

Séaghdha, D. O. (2010). Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444.

Shapira, B., Shoval, P., and Hanani, U. (1997). Stereotypes in information filtering systems. *Information Processing & Management*, 33(3):273–287.

Shaw, J. A. and Fox, E. A. (1994). Combination of multiple searches. In *Proceedings of The Third Text REtrieval Conference, TREC 1994*, pages 105–108.

Shen, X., Tan, B., and Zhai, C. (2005). 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.

Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 101–110.

Sheng, E., Chang, K.-W., Natarajan, P., and Peng, N. (2019). The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3407–3412.

Shih, Y.-S. (1999). Families of splitting criteria for classification trees. *Statistics and Computing*, 9(4):309–315.

Shokouhi, M., White, R. W., Bennett, P., and Radlinski, F. (2013). Fighting search engine amnesia: Reranking repeated results. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 273–282.

Sieg, A., Mobasher, B., and Burke, R. (2007). Web search personalization with ontological user profiles. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 525–534.

Silverstein, C., Marais, H., Henzinger, M., and Moricz, M. (1999). Analysis of a very large web search engine query log. In *ACm SIGIR Forum*, volume 33, pages 6–12. ACM New York, NY, USA.

Silvestri, F. (2010). Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1–2):1–174.

Singer, G., Norbisrath, U., and Lewandowski, D. (2012). Ordinary search engine users assessing difficulty, effort, and outcome for simple and complex search tasks. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 110–119.

Sloan, M., Yang, H., and Wang, J. (2015). A term-based methodology for query reformulation understanding. *Information Retrieval Journal*, 18(2):145–165.

Smith, C. L. and Kantor, P. B. (2008). User adaptation: good results from poor systems. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 147–154.

Smyth, B. (2007). A community-based approach to personalizing web search. *Computer*, 40(8):42–50.

Song, R., Luo, Z., Nie, J.-Y., Yu, Y., and Hon, H.-W. (2009). Identification of ambiguous queries in web search. *Information Processing & Management*, 45(2):216–229.

Song, R., Zhang, M., Sakai, T., Kato, M. P., Liu, Y., Sugimoto, M., Wang, Q., and Orii, N. (2011). Overview of the ntcir-9 intent task. In *NTCIR*.

Sontag, D., Collins-Thompson, K., Bennett, P. N., White, R. W., Dumais, S., and Billerbeck, B. (2012). Probabilistic models for personalizing web search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 433–442.

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., and Nie, J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562.

Spärck-Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Spärck-Jones, K., Robertson, S. E., and Sanderson, M. (2007). Ambiguous requests: implications for retrieval tests, systems and theories. In *ACM SIGIR Forum*, volume 41, pages 8–17. ACM New York, NY, USA.

Spärck-Jones, K., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information processing & management*, 36(6):779–808.

Spink, A., Wolfram, D., Jansen, M. B., and Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440.

Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684.

Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., and Chen, Z. (2005). Cubesvd: a novel approach to personalized web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 382–390.

Tang, Z. and Yang, G. H. (2017). Investigating per topic upper bound for session search evaluation. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 185–192.

Teevan, J., Adar, E., Jones, R., and Potts, M. A. (2007a). Information re-retrieval: repeat queries in yahoo's logs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 151–158.

Teevan, J., Dumais, S. T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456.

Teevan, J., Dumais, S. T., and Horvitz, E. (2007b). Characterizing the value of personalizing search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 757–758.

Teevan, J., Dumais, S. T., and Horvitz, E. (2010). Potential for personalization. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(1):1–31.

Teevan, J., Morris, M. R., and Bush, S. (2009). Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 15–24.

The Lemur project (2009). Clueweb09 related data. `https://www.lemurproject.org/clueweb09/related-data.php`. Online.

The Lemur project (2012). Clueweb12 related data. `https://www.lemurproject.org/clueweb12/related-data.php`. Online.

Thomas, P. and Hawking, D. (2006). Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 94–101.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Tyler, S. K. and Teevan, J. (2010). Large scale query log analysis of re-finding. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 191–200.

Ustinovskiy, Y. and Serdyukov, P. (2013). Personalization of web-search using short-term browsing context. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1979–1988.

Van Gysel, C., Kanoulas, E., and de Rijke, M. (2016). Lexical query modeling in session search. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 69–72.

van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworth.

Voorhees, E. M. (2001). The philosophy of information retrieval evaluation. In *Workshop of the cross-language evaluation forum for european languages*, pages 355–370. Springer.

Vu, T., Nguyen, D. Q., Johnson, M., Song, D., and Willis, A. (2017). Search personalization with embeddings. In *European Conference on Information Retrieval*, pages 598–604. Springer.

Vu, T., Willis, A., Tran, S. N., and Song, D. (2015). Temporal latent topic user profiles for search personalisation. In *European Conference on Information Retrieval*, pages 605–616. Springer.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112.

Wang, B., Liu, K., and Zhao, J. (2016). Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1288–1297.

Wang, C., Liu, Y., Zhang, M., Ma, S., Zheng, M., Qian, J., and Zhang, K. (2013). Incorporating vertical results into search click models. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 503–512.

Wang, Y. and Agichtein, E. (2010). Query ambiguity revisited: clickthrough measures for distinguishing informational and ambiguous queries. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 361–364.

Wasserman, S. and Faust, K. (2009). *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press.

Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185.

White, R. (2013). Beliefs and biases in web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12.

White, R. W. (2016). *Interactions with search systems*. Cambridge University Press.

White, R. W., Bennett, P. N., and Dumais, S. T. (2010). 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.

White, R. W., Bilenko, M., and Cucerzan, S. (2007). Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 159–166.

Wolfram, D., Spink, A., Jansen, B. J., Saracevic, T., et al. (2001). Vox populi: The public searching of the web. *Journal of the association for information science and technology*, 52(12):1073–1074.

Wu, Q., Burges, C. J., Svore, K. M., and Gao, J. (2010). Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270.

Xia, F., Liu, T.-Y., Wang, J., Zhang, W., and Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.

Xia, L., Xu, J., Lan, Y., Guo, J., and Cheng, X. (2015). Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 113–122.

Xia, L., Xu, J., Lan, Y., Guo, J., Zeng, W., and Cheng, X. (2017). Adapting markov decision process for search result diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 535–544.

Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., and Li, H. (2010). 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.

Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

Xu, J. and Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398.

Xue, G.-R., Han, J., Yu, Y., and Yang, Q. (2009). User language model for collaborative personalized search. *ACM Transactions on Information Systems (TOIS)*, 27(2):1–28.

Xue, Y., Cui, G., Yu, X., Liu, Y., and Cheng, X. (2014). Ictnet at session track trec2014. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC.*

Yang, G. H., Dong, X., Luo, J., and Zhang, S. (2018). Session search modeling by partially observable markov decision process. *Information Retrieval Journal*, 21(1):56–80.

Yang, G. H., Tang, Z., and Soboroff, I. (2017). TREC 2017 dynamic domain track overview. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC*.

Yang, Y. and Lad, A. (2009). Modeling expected utility of multi-session information distillation. In *Conference on the Theory of Information Retrieval*, pages 164–175. Springer.

Yin, D., Xue, Z., Qi, X., and Davison, B. D. (2009). Diversifying search results with popular subtopics. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009*.

Yue, Y. and Joachims, T. (2008). Predicting diverse subsets using structural svms. In *Proceedings of the 25th international conference on Machine learning*, pages 1224–1231.

Yue, Y., Patel, R., and Roehrig, H. (2010). Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*, pages 1011–1018.

Zamani, H. and Croft, W. B. (2017). Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 505–514.

Zamir, O. and Etzioni, O. (1999). Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11-16):1361–1374.

Zhai, C., Cohen, W. W., and Lafferty, J. D. (2003). Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17. ACM.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., and Ma, W.-Y. (2005). Improving web search results using affinity graph. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511.

Zhang, P., Li, J., Wang, B., Zhao, X., Song, D., Hou, Y., and Melucci, M. (2016a). A quantum query expansion approach for session search. *Entropy*, 18(4):146.

Zhang, Y. and Moffat, A. (2006). Some observations on user search behaviour. *Aust. J. Intell. Inf. Process. Syst.*, 9(2):1–8.

Zhang, Z., Wang, J., Wu, T., Ren, P., Chen, Z., and Si, L. (2016b). Supervised local contexts aggregation for effective session search. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR*, pages 58–71.

Zhao, L. and Callan, J. (2010). Term necessity prediction. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 259–268.

Zhao, X. W., Guo, Y., He, Y., Jiang, H., Wu, Y., and Li, X. (2014). We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1935–1944.

Zheng, G. and Callan, J. (2015). 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.

Zheng, W., Wang, X., Fang, H., and Cheng, H. (2012). Coverage-based search result diversification. *Information Retrieval*, 15(5):433–457.

Zhou, Y., Dou, Z., and Wen, J. (2020). Encoding history with context-aware representation learning for personalized search. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, pages 1111–1120. ACM.

Zuccon, G. and Azzopardi, L. (2010). Using the quantum probability ranking principle to rank interdependent documents. In *European Conference on Information Retrieval*, pages 357–369. Springer.