Predicting Search Intent Based on Pre-Search Context

Weize Kong[1], Rui Li[2], Luo Jie[2], Aston Zhang[3], Yi Chang[2], James Allan[1]
[1]Center for Intelligent Information Retrieval, University of Massachusetts Amherst, Amherst, MA 01003
[2]Yahoo! Labs, 701 First Ave, Sunnyvale, CA 94089
[3]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801
{wkong,allan}@cs.umass.edu, {ruililab,luojie,yichang}@yahoo-inc.com, lzhang74@illinois.edu

ABSTRACT

While many studies have been conducted on query understanding, there is limited understanding on why users start searches and how to predict search intent. In this paper, we propose to study this important but less explored problem. Our key intuition is that searches are triggered by different pre-search contexts, but the triggering relations are often hidden. For example, a user may search "bitcoin" because of a news article or an email the user just read, but the system does not know which of the pre-search contexts (the news article or the email) is the triggering source. Following this intuition, we conduct an in-depth analysis of pre-search context on a large-scale user log, which not only verifies the hidden triggering relations in the real world but also identifies a set of important characteristics of pre-search context and their triggered queries. Since the hidden triggering relations make it challenging to directly use pre-search context for intent prediction, we develop a mixture generative model to learn without any supervision how queries are triggered by different types of pre-search context. Further, we discuss how to apply our model to improve query prediction and query auto-completion. Our experiments on a large-scale of real-world data show that our model could accurately predict user search intent with pre-search context and improve upon the state-of-the-art methods significantly.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Search Process

Keywords: Search Context; Pre-Search Context; Search Intent; Query Auto-Completion; Query Prediction

1. INTRODUCTION

To improve user search experience, many studies have been proposed to understand user search intent given issued queries (*e.g.*, query suggestion and query disambiguation) via exploring clickthrough data, user search history, and search session context. However, there is very limited work on search intent prediction – *predicting what a user is going to search even before the search task starts.*

Search intent prediction is an important problem, as it will largely improve search experience. First, it could enable "query-less" search (*e.g.*, Google Now) to greatly save user efforts. For example, if we could predict that a user is looking for Philippine Peso's value, we could directly recommend related queries (*e.g.*, "Philippine Peso") or their results once the user opens a mobile search app, without any user input. In addition, it could also improve existing applications such as query auto-completion and query disambiguation. For example, when the user types "P" in a search box, we can accurately suggest query "Peso" or "Philippine Peso" instead of "Pinterest" or "Paypal", which are more likely to be suggested by current search engines. When the user searches "Peso" as his query, we could correctly interpret the intent as Philippine Peso instead of Mexico Peso.

Recently Cheng et al. [7] found that many searches are triggered by webpages users browsed, and proposed a model to predict queries triggered by a given browsed webpage. While this work provides a good start for search intent prediction, it has critical limitations in reality. In the real-world setting, searches could also be triggered by factors other than browsing, and systems often do not know if the current search will be triggered by the browsed webpage or other factors. Therefore, blindly predicting search intent solely based on browsed webpage is inadequate in practice. In this paper, we study this important problem of predicting search intent in this more realistic setting.

A Key Insight We hypothesize that searches are triggered by different contexts prior to the search activities, which we call *pre-search contexts*. For example, a user searches "peso", because she reads a news article about "Devaluation of Philippine peso" and she is interested in its current value; a user searches "restaurants nearby" as she is visiting a new place and want to taste local food; and a user searches "Lady Gaga tour" as she hears about Lady Gaga's coming concert and wants to book tickets. In these examples, the news article, the new location, and the coming event are the pre-search contexts that trigger users to search. It is clear that pre-search context is very different from user search history or search session context, which are explored by many previous studies for understanding search intent.

We emphasize that a pre-search context, by definition, is just prior to the search but does not necessarily trigger it. This definition reflects the hidden nature of triggering relations between pre-search context and searches in *a real-world setting*. That is, while many types of pre-search con-

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

ACM 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767757.

text (*e.g.*, news articles, new locations, events) could trigger searches, the system does not observe which particular pre-search context triggers the current search. Ideally, if we can capture various types of pre-search context, and identify the particular one that triggers searches each time, we could predict search intent based on the identified pre-search context.

An In-depth Analysis In Section 3 we conduct an in-depth analysis to understand how pre-search context triggers queries. We focus on a specific type of pre-search context – the news article a user browsed before the search, because: 1) a large number of search queries are triggered by news pre-search context daily, as we will show in the analysis; 2) it is easier to log browsing activity than other off-line activities in practice; 3) studying news pre-search context will provide guidelines for exploring other pre-search context in the future.

In the analysis, we use large-scale real-world data by joining user browsing logs (*i.e.*, what news articles users have read) and search logs (*i.e.*, what queries users have searched after reading news) from Yahoo!. We first find that a significant amount of queries are triggered by news pre-search context (*i.e.*, tens of millions queries per day), which verifies the assumption of queries being triggered by pre-search context. More importantly, we discover insightful characteristics of news pre-search context and their triggered queries. Some of our interesting observations are: 1) news articles often trigger *new* queries (*i.e.*, 96% of triggered queries have never been searched by their users before), which are often difficult to predict or understand if only using history data; 2) most of the triggered queries are related to named entities mentioned in the news articles; 3) triggered queries are more likely to appear in the beginning part of those articles; 4) triggered queries could diverge from the main topics of the news articles, which are quite different from queries that lead clicks to the articles.

While this analysis shows the great potential of using news pre-search context for predicting search intent, it also validates the hidden nature of triggering relations in the real-world setting as mentioned previously. Based on the data, we estimate a large percent of queries are triggered by other pre-search context instead of news articles. The system does not know if current search will be triggered from the browsed news articles or other pre-search context. This hidden nature poses two challenges in reality that, to the best of our knowledge, have not been addressed before: 1) how to predict search intent with hidden triggering relations; 2) if/how can we learn prediction models and avoid using expensive labels for uncovering the hidden triggering relations (*i.e.*, identify which pre-search context is the triggering factor).

An Intent Prediction Model To address these real-world challenges, in Section 4, we develop a pre-context aware search intent model that learns how queries are triggered by different pre-search context with hidden triggering relations without any supervision. More specifically, we first introduce the model as a general generative mixture model to principally captures that a search intent could be triggered by different pre-search context with unknown triggering source. We then customize the expectation-maximization (EM) algorithm to learn the parameters of our model, so that we can take advantage of large-size browsing and search logs that are directly available, avoiding relying on expensive manually labeled data. In addition, we discuss how to instantiate our model for query prediction and query auto-completion tasks as concrete examples to validate the generality of our proposed model. We also identify a set of effective features for our model based on our analysis in Section 3.

Comprehensive Experiments Finally, in Section 5, we carry out a comprehensive set of experiments to evaluate our model on query prediction and query suggestion tasks. We show that: 1) pre-search context is very useful in predicting queries compared to other types of data (*e.g.*, user search history and clickthrough data); 2) the state-of-the-art method [7] is inadequate in a real-world setting; 3) instead, our model captures pre-search context with hidden triggering relations effectively and outperforms the state-of-the-art method (*i.e.*, 25% improvement) under the real-world setting; 4) our model also *consistently* improves over other methods in query auto-completion, and achieves larger improvement in the most challenging cases, in which users have only typed one or two letters.

2. RELATED WORK

We divide information used for modeling user search intents into two categories – long-term history and short-term context.

The *long-term history* contains user behavior information, such as queries, clickthrough and browsed webpages, over a long period. It is often used to build user profiles and capture users' general search interest. For example, in query auto-completion, many search engines suggest the completions that have been most popular among users in the past history [1]. For personalized search, Gauch et al. [12] learn user profiles from browsing history, Speretta and Gauch [25] build profiles using search history. The *short-term context* instead provides a more direct information for inferring users' current search intent. We believe there exist two types of short-term contexts – pre-search context and in-search context. The *pre-search context*, as we defined, is the search context that is prior to a search task and could trigger the search; in-search context is the search context during a search task, such as query reformulation and user clickthrough during a search session.

In-Search context. There is a large body of work studying in-search context. In almost all of the work, in-search context is essentially used as additional information for understanding search intent during a search task. Different models have been proposed for utilizing in-session context to improve various aspects of search, including query classification [28, 27], query suggestions and auto-completion [5, 6, 4, 1, 11], document ranking [23, 29, 17], modeling search satisfaction [14, 13, 24, 22], search evaluation at session or task level [16, 15], and search personalization [26, 18, 2]. Despite the rich types of contexts and applications covered by these studies, the kind of context that could trigger a search task (*i.e.*, pre-seach context) is still relatively unexplored.

Pre-Search context. There is limited work studying pre-search context. Dumais et al. [10] have a demonstration that automatically suggests queries while users are reading emails. However, details of the model used for the system are not provided, and no evaluation is performed for the system. Rahurkar et al. [20] detect if a query searched right after browsing is relevant to the browsed webpage, and achieve

nearly perfect classification performance with 0.96 precision and 0.90 recall. But they do not consider modeling intent based on context. The same authors also carry out a preliminary study [21] that uses the browsed webpage to improve search relevance. The work aims to find webpages similar to the browsed webpage, which may not be what the user needs: as we will show that triggered search intent may diverge from the original browsed webpage in Section 3.3.1. Liebling et al. [19] try to predict what webpages users are likely to look for after browsing a webpage, based on the popularity of webpages that have been searched under the same browsing context. However, their approach is difficult to be generalized for fresh or less-popular webpages that do not have enough browsing history.

There are two recent studies [7, 3] that are similar to our work. Both of them try to predict/suggest what users are likely to search about a browsed webpage. Cheng et al. [7] collect searches after users browse a webpage in the search history, and then rank them as suggestions for that webpage using a learning to rank framework. Instead of relying on historic browsing and search information, Bordino et al. [3] focus on using webpage content, so that their approach can be generalized for previous unseen webpages. Our work significantly differs from the two studies in that: 1) we study a different problem. Motivated by the hidden nature of triggering relations in the real-world setting, we target at predicting search intent based on pre-search context, which may or may not trigger the current search, while they target at predicting search intent for a given *known* triggering factor, which is normally unrealistic; 2) we perform an in-depth analysis. It verifies the hidden nature of triggering relations and discovers insightful characteristics of pre-search context, some of which have important implications on feature design. Previous analysis [7] is limited to verifying triggering relations ; 3) our model captures hidden triggering relations and learns from large-size "free" unlabeled data, while their models require expensive human labels to identify triggering sources for learning.

3. ANALYSIS OF PRE-SEARCH CONTEXT

In this section, we present an in-depth analysis of how pre-search context triggers users to search, focusing on browsed news articles (as explained in Section 1). Particularly, we first verify the triggering relations and their hidden nature in a real-world setting, and then explore the characteristics of news pre-search context and its triggered queries.

3.1 Dataset

To understand how pre-search context triggers queries in real world, we join *browsing logs*, which record user "browsing events" on Yahoo! news sites (*e.g.*, News, Sports, Finance), and *search logs*, which record user "search events" on Yahoo! Search, and then focus on "browsing-search pairs" in the joined logs. All the logs are from the English/US market and are anonymized. Formally, we can abstract our logs as follows.

- A **browsing event** contains three fields {*userID*, *timestamp*, *URL*}, which records the event that a user browses a URL at a specific time;
- A **search event** contains three fields {*userID*, *timestamp*, *query*}, which records the event that a user issues a query at a specific time.
- A **browsing-search pair** is a pair of a browsing event and a subsequent search event within a predefined time window (*e.g.*, 30 mins) in the logs.

In the rest of the paper, we refer to the webpage and the query in a browsing-search pair as **browsed webpage** and **following query** respectively. As not all the following queries are indeed triggered by the browsed webpages, we name the following queries that are triggered by the browsed webpages as **triggered queries** and the other following queries as **non-triggered queries**.

We process 25 successive days of browsing and search logs in February 2014. In the following studies and experiments, we use the logs from the first 24 days as *history data*, from which we extract statistics such as user search history and the queries that lead clicks to a certain webpage, and the logs from the last day as *experiment data*, which we use for our analysis and experiments. We have 1,796,313 browsing-search pairs after pre-processing and filtering (*e.g.*, filtering out browsing-search pairs whose webpages are not accessible due to outdated links or technical issues). We randomly sample 6,000 pairs for annotators to label. The annotators are asked to label whether or not the following query is triggered by the browsed webpage or whether they cannot decide. Those labeled pairs are further split into 3 different sets randomly, namely STUDY, TRAIN and EVALUATE. We use both STUDY and TRAIN for our analysis, TRAIN for model training, and EVALUATE for testing. Table 1 shows the statistics of annotated pairs in the three sets.

Table 1: Label distributions of three datasets.

	STUDY	TRAIN	EVALUATE
triggered	126 (6.3%)	114 (5.7%)	106 (5.3%)
non-triggered	1,849 (92.5%)	1,878 (93.9%)	1,891 (94.6%)
cannot decide	25 (1.3%)	8 (0.4%)	3 (0.2%)

3.2 Verifying Hidden Triggering Relations

To empirically justify our key insight, we first need to verify the triggering relations (*i.e.*, whether queries are triggered by news pre-search context) in our user logs. Particularly, by investigating the logs in our dataset, we first find that 3.6% of all the webpage browsing events are followed by immediate search events (no other events in between browsing and search) within 30 minutes. These "following" queries consist of 10% of all search traffic. Further, from the human-annotated browsing-search pairs, shown in Table 1, we see that around 6% of these "following" queries are triggered by their browsed webpages. Thus, we estimate at least 0.6% all search queries are triggered by news pre-search context (*i.e.*, browsed news articles). This actually underestimates the significance of news triggered searches due to that users may search in other search engines after browsing, which is not logged in our data. If we consider that there are many billions of queries everyday [1], tens of millions of queries are triggered by news pre-search context. As an illustration, Table 2 shows some following queries, comprising triggered queries (queries triggered by the browsed webpages) and non-triggered queries for three webpages. We can clearly see that some following queries (*e.g.*, "what is a bitcoin") indeed are triggered by the corresponding browsed webpages (*e.g.*, Bitcoin exchange Mt. Gox goes dark in blow to virtual currency).

[1]Google processes 3.5 billion searches daily `http://www.statisticbrain.com/google-searches/`

Table 2: Examples for the triggered, non-triggered and leading queries from three popular news articles.

Webpage A: Bitcoin exchange Mt. Gox goes dark in blow to virtual currency	
`http://finance.yahoo.com/news/bitcoin-exchange-mt-goxs-website-062314988.html`	
triggered	bitcoin, what is bitcoin, bitcoin heist
non-triggered	facebook, amanda knox, galaxy s5 launch
Webpage B: Harold Ramis Star of Ghostbusters Director of Caddyshack Dies at 69	
`https://movies.yahoo.com/blogs/yahoo-movies/-ghostbusters--actor-harold-ramis-dies-at-69-173530161.html`	
triggered	harold ramis, animal house movie, autoimmune inflammatory vasculitis, cast of ghostbusters
non-triggered	craigslist, amanda bynes, ali fedotowsky, ebay
leading	harold ramis death, harold ramis died, ghostbuster movie actor dies, what star of ghostbuster movie dies
Webpage C: 6 Stars Who Have Returned to Regular Jobs	
`https://celebrity.yahoo.com/blogs/yahoo-celebrity/6-stars-returned-regular-jobs-194630934.html`	
triggered	susan boyle, susan boyle bio, susan boyle movie, steven seaga
non-triggered	facebook, free credit score
leading	regular jobs for celebrities, stars that returned back to regular jobs

The analysis above verifies the triggering relations between news pre-search context and searches. However, the triggering relations are hidden: the system does not directly observe whether queries will be triggered by news pre-search context or other pre-search context (*e.g.*, locations, events). Based on Table 1, we estimate that around 94% of following queries are triggered by pre-search context other than browsed news articles, and the system does not know which pre-search context triggers the current search. This hidden nature poses two real-world challenges that are not addressed in previous studies [7, 3]: 1) how to predict searches with systems not knowing when/if searches will be triggered by the browsed webpages; 2) if/how can we learn prediction models and avoid using expensive labels for identifying which queries are triggered by the browsed webpages. We will propose a generative mixture model to principally address them in Section 4.

3.3 Characterizing Triggered Queries

Next, to predict user search intent based on pre-search context, we need to characterize how pre-search context triggers searches. Particularly, we derive characteristics of triggered queries, which distinguish them from other following queries, from two main aspects: 1) relevance between browsed webpages and following queries and 2) types of following queries themselves.

3.3.1 Relevance between Following Queries and Pre-Search Context

As it is natural to expect that triggered queries are relevant to the content of their browsed webpages (the page that triggered the query), we first focus on whether triggered/non-triggered queries are actually relevant to the browsed webpages.

Text Matching We start with a basic question – *can we distinguish triggered queries and non-triggered queries by how well they match the text content of browsed webpages*?

To answer this question, we compare triggered queries with non-triggered queries according to two text matching metrics: 1) *exactly matching*, which measures whether a query appears in a webpage; 2) *overlapping*, which measures whether a query has at least one word overlapping with the webpage, ignoring stopwords in the query. In Table 3, we show the percentages of the matched queries and the overlapped queries among the triggered/non-triggered queries, as well as the percentages of the triggered/non-triggered queries belong to the matched and overlapped queries. We have two important observations from the results.

First, the results confirm our expectation that triggered queries are more relevant to browsed webpages than the non-triggered ones. Particularly, 48.41% (96.03%) of triggered queries match (overlap) with browsed webpages, while only 0.49% (8.22%) of non-triggered queries do. In fact, by examining queries in detail, we find that 1) less than 4% of triggered queries that do not overlap browsed webpages are semantically related to them (*e.g.*, after browsing a webpage about Ford, a user searches a specific automobile store, whose name does not appear in the webpage) and the 0.5% of the non-triggered queries which exactly match browsed webpages, are mostly popular navigational queries (*e.g.*, "facebook", "google" and "espn").

Second, the results suggest that, unfortunately, we cannot distinguish triggered from non-triggered queries very well with text matching only. Specifically, we can only obtain a high precision (87.14%) but low recall (48.41%) classifier or a high recall (96.03%) but low precision (44.32%) classifier for predicting triggered/non-triggered queries with the matching or overlapping measures respectively. This implies that we should also consider other features for modeling triggering relations.

Table 3: Matching and overlapping of triggered and non-triggered queries.

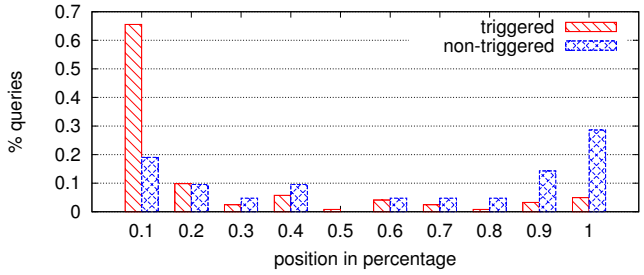
	% exactly matching	% overlapping
triggered	48.41%	96.03%
non-triggered	0.49%	8.22%
	% triggered	% non-triggered
exactly matching	87.14%	12.86%
overlapping	44.32%	56.68%

Matching Positions Due to the limitations of matching and overlapping in distinguishing triggered and non-triggered queries, we next investigate whether *where the matched queries of triggered/non triggered queries appear in the webpages* can distinguish the classes. Our intuition is that because users read from top to bottom and news articles often position important/interesting information at the top, triggered queries are likely to come from the beginning of browsed webpages.

To answer the question, we plot distributions of the document positions where triggered/non-triggered queries first

appear in Figure 1. In the figure, document positions from top to bottom are normalized to a value between 0 and 1 by word number. Triggered/non-triggered queries that do not appear in the news articles have no valid document positions, and are thus not included in this plot. Figure 1 shows that most triggered queries appear in the beginning part of the news articles, and the non-triggered queries randomly appear in the beginning and the end of browsed webpages. The results here verify our intuition above, which could be an effective feature for distinguishing triggered and non-triggered queries.

Figure 1: Distributions of triggered & non-triggered queries' first occurrences in their browsed webpages.



Topic Relevance After measuring text similarity, we further measure *how relevant the triggered and non-triggered queries of a webpage are to its main topic.*

To answer this question, we compare triggered and non-triggered queries with the title of a webpage, which is a good representation of the main topic of the webpage. Specifically, we measure the cosine similarity between queries and news article titles based on their term-frequency vectors. In addition, we compute the similarity between "leading" queries of a webpage and the title of the webpage, as a baseline. Here, leading queries of a webpage are queries after which people click on the webpage, known to be relevant to the main topic of the webpage [8, 30]. Table 2 also shows some leading queries of two of the webpages.

Table 4 shows that, as we expected, triggered queries are more relevant to the topic of a browsed webpage than the non-triggered ones. However, interestingly, both triggered and non-triggered queries are much less relevant to the main topics of browsed webpages than the leading queries. This can also be illustrated by the examples in Table 2. For Webpage B, which is about the death of a film director, the leading queries all focus on the death of Harold Ramis, while triggered queries might be about his movies, "animal house movie". It can be explained as users may be interested in any specific information in a webpage.

Table 4: Cosine similarity between triggered/non-triggered/leading queries and news titles.

Triggered	Non-Triggerd	Leading
0.1641	0.0274	0.4782

While the above results suggest relevance to the main topic could be a marginally effective feature, the results are still valuable to us because: 1) they reveal an important fact that the triggered queries of a webpage could diverge from its main topic; and 2) they show clear differences between leading queries and triggered queries, which motivate of our study of triggered queries from a new aspect.

3.3.2 *Types of Following Queries*

We would like to understand if some particular types of queries are more likely to be triggered by a *news* pre-search context. Intuitively, we assume that if users issue queries because of reading articles (*e.g.*, Bitcoin exchange Mt. Gox goes dark in blow to virtual currency), they are likely to look for explanations of some unknown concepts (*e.g.*, "what is bitcoin") or related information for some entities (*e.g.*, "bitcoin value"). Thus we expect the triggered queries are *new* to the user and *entity oriented*. Next, we empirically verify these two assumptions.

New assumption We verify that triggered queries are more likely to be new to the users (who issued the queries) than non-triggered ones by checking if triggered queries have been searched by the users before. We assume that a query never searched by a user before is likely to be a "new" concept to the user. We find almost all (95.4%) of news-triggered queries have never been searched by their users (the particular user who issued the triggered query) in their personal history data (previous 24 days), while this percentage is much lower for non-triggered queries (78.64%) and for all queries issued by the same set of users in the same day (77.2%). This assumption can also be illustrated from the examples in Table 2, where in the first example some query prefixes start with "what is", suggesting users are searching concepts new to them.

This result not only reveals another important characteristic of news triggered queries, but also implies that they will be much more difficult to predict than other queries if we solely depend on user search history.

Entity oriented assumption To verify that triggered queries are more likely to be about entities mentioned in browsed webpages than are non-triggered ones, we apply an in-house named entity extractor on browsed webpages, and compare triggered/non-triggered queries with the recognized named entities. We find that over half (55.8%) of the triggered queries contain at least one named entity recognized in the browsed webpages, and the percentage goes down to 1% for the non-triggered queries. Among the triggered queries with entities, 61.2% are exactly named entities without any other modifiers: for example, after reading Webpage C in Table 2, many users search query "susan boyle", who is a celebrity mentioned in the webpage. Another 38.8% of these queries are named entity with some modifiers: for example, users also issue the query "susan boyle bio" and "susan boyle movie" after reading that news page. This observation clearly suggests that the named entities mentioned in the browsed webpages will be useful in predicting search intent triggered by the webpages.

We note that there could be more named entity related queries than these estimations, because 1) many named entities in the news articles are not recognized by our entity extractor, e.g. "bitcoin", and 2) queries may use different variations of the named entity: *e.g.*, "bit coin" and "bitcoin", "the walking dead" and "walking dead".

4. PRE-SEARCH CONTEXT AWARE SEARCH INTENT MODELING

Though our analysis shows news pre-search context provides useful information for search intent prediction, it also

recognizes the hidden triggering relation challenge in real systems, that has not been addressed previously (Section 3.2). To address that challenge, we propose a Pre-search Context-aware Intent Model (PCIM) based on a generative mixture model to effectively learn from a large amount of unlabeled data. This model differs from a previous model [7] in that 1) it captures multiple types of pre-search context with *unknown* triggering source; and 2) it takes advantage of a large amount of unlabeled data to learn without supervision.

With the hope to improve different search related tasks, we first describe PCIM as a general search intent model. Then we illustrate how this general model can be applied to two specific applications, query prediction and query auto-completion.

4.1 Pre-Search Context Aware Intent Model

4.1.1 Search Intent

We define a search intent model as a probability distribution over a set of intents, $P(I), \forall I \in \mathcal{I}$, where the set of intents $\mathcal{I}$ can be composed of words, phrases, or even ODP categories, depending on the specific application. For example, in query prediction and query auto-completion, the set of intents contains all the query candidates that the user may search. Instead, if we use ODP categories as intents, then the search intent model can also be applied to classify search intent. We use this general definition for our search intent model, so that the model can be applied to different search related tasks. We will apply this general model to query prediction and query auto-completion as examples in Section 4.2.

4.1.2 A Generative Mixture Model

As discussed in Section 1, search intents can be triggered by different types of pre-search contexts C, such as browsed webpages ("Devaluation of Philippine Peso"), locations (new place visited) and events (Lady Gaga's coming concert). We also denote the set of all the pre-search context as $\mathcal{C}$. Based on this, we assume that search intents are generated from different pre-search contexts according to the following generative process:

1. A pre-search context C is selected as the triggering source from an unknown multinomial distribution, $C \sim$ *Multinomial*($\boldsymbol{\lambda}$), where $C \in \mathcal{C}$.
2. The search intent I is generated by the given pre-search context C according to another unknown distribution $I \sim P(I|C)$.

According to this generative process, the mixture generative model for I can be written as

$$P(I) = \sum_{C \in \mathcal{C}} P(I|C)P(C), \tag{1}$$

where $P(C) = \lambda_C$ is the prior for each pre-search context following the multinomial distribution, *Multinomial*($\boldsymbol{\lambda}$). While there are many different ways to model the intent distribution $P(I|C)$, we choose a standard log-linear model as the hypothesis function. Its function can be written as

$$P(I|C) = \frac{1}{Z} \exp\left(\boldsymbol{w}_C^T \boldsymbol{f}(I, C)\right), \tag{2}$$

where $Z = \sum_{I \in \mathcal{I}} \exp\left(\boldsymbol{w}_C^T \boldsymbol{f}(I, C)\right)$ is the normalizer, $\boldsymbol{f}(I, C)$ is a vector of features characterizing how likely will intent I be triggered by pre-search context C, and $\boldsymbol{w}_C$ are the corresponding feature weights. The details of used features are described in Section 4.2.3.

4.1.3 Parameter Estimation

The PCIM model defined in Equation 1 contains parameters $\boldsymbol{\theta} = \{\boldsymbol{\lambda}, \boldsymbol{W}\}$, where $\boldsymbol{\lambda} = \{\lambda_C\}$ is the prior probabilities for each type of pre-search context; $\boldsymbol{W} = \{\boldsymbol{w}_C\}$ are the feature weights for each type of pre-search context in calculating $P(I|C)$. To estimate $\boldsymbol{\theta}$, a straightforward way is to maximize the likelihood value of $P(\{I_i\}_i)$ on a given training data set $\{I_i\}_{i=1,\cdots,N}$, where N is the total number of training instances. Assuming independence between different search intent examples I_i, the log-likehood of the example collection is

$$l(\boldsymbol{\theta}) = \log \prod_i P(I_i; \boldsymbol{\theta}) = \sum_i \log \sum_{C \in \mathcal{C}} P(I_i|C; \boldsymbol{w}_C) P(C; \lambda_C). \tag{3}$$

Unfortunately, since log-likelihood in Equation 3 involves logarithm of a summation, there is no exact analytical solution for $\boldsymbol{\theta}$. Therefore, we propose to use the EM algorithm for estimating $\boldsymbol{\theta}$ instead.

In the E-step, we evaluate the posterior probability of a search intent being generated from a particular pre-search context given the previous model parameter estimation $\boldsymbol{\theta}' = \{\boldsymbol{\lambda}', \boldsymbol{W}'\}$:

$$P(C|I_i; \boldsymbol{\theta}') \propto P(I_i|C; \boldsymbol{w}'_C) P(C; \lambda'_C). \tag{4}$$

In the M-Step, based on posterior estimates from the E-step, we maximize the lower bound of the log-likelihood:

$$l(\boldsymbol{\theta}) \geqslant l'(\boldsymbol{\theta}) = \sum_i \sum_{C \in \mathcal{C}} P(C|I_i; \boldsymbol{\theta}') \log \frac{P(I_i|C; \boldsymbol{w}_C) P(C; \boldsymbol{\lambda}_C)}{P(C|I_i; \boldsymbol{\theta}')}. \tag{5}$$

By taking derivative of $l'(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\lambda}$ and setting the derivative to zero, we can easily update $\boldsymbol{\lambda}$ by $\lambda_C = \frac{P(C|I_i;\boldsymbol{\theta}')}{N}$. For $\boldsymbol{W}$, we calculate the derivative according to the following function, and use gradient ascent to maximize $l'(\theta)$,

$$\nabla_{\boldsymbol{w}_C} l'(\boldsymbol{\theta}) = \sum_i P(C|I_i; \boldsymbol{\theta}') \left(\boldsymbol{f}(I_i, C) - \frac{\sum_{I' \in \mathcal{I}_i} \exp\left(\boldsymbol{w}_C^T \boldsymbol{f}(I', C)\right) \boldsymbol{f}(I_i, C)}{\sum_{I' \in \mathcal{I}_i} \exp\left(\boldsymbol{w}_C^T \boldsymbol{f}(I', C)\right)} \right), \tag{6}$$

where $\mathcal{I}_i$ is the set of candidate search intents for the current search intent I_i.

4.2 Applications

The PCIM described above is a general model that could be applied to different search related tasks. Now we show how to apply it to query prediction and query auto-completion as examples. In the query prediction task, our goal is to suggest a ranked list of queries that a user is likely to search (after the user browses a webpage, which may or may not trigger the search in the real-world setting). The query auto-completion task is similar to query prediction, except that it predicts (auto-complete) the query after the user has typed some initial characters of the query, which restrict the set of possible suggested queries to those that begin with the initial letters as prefix.

To apply PCIM to the two applications, we need to 1) specify search intent and pre-search context representations, 2) adapt the model for the chosen representation, and 3) design specialized features for the specific applications. Next, we describe each of these issues in detail.

4.2.1 Representation

In both query prediction and query auto-completion tasks, the basic unit of the search intent set $\mathcal{I}$ is a query. Hence we will use query Q instead of intent I in our specific models. The candidate queries can be pooled from search logs and the browsed webpages. For query auto-completion, those candidate queries need to begin with the initial letters the user has typed. As an illustration, we consider only two types of pre-search context C in both tasks for the sake of feasibility and simplicity: 1) the webpage D the user just browsed before search; 2) a special background pre-search context G based on long-term search history, which we assume summarizes all other uncaptured pre-search context. We use this background pre-search context, because many other types of pre-search context could trigger searches, but are not accessible to the system (*e.g.*, locations, events). We believe it is easy to extend our model for more types of pre-search context when they are accessible, which we leave as future work.

4.2.2 Model

Given the representation, PCIM in Equation 1 can be rewritten as

$$P(Q) = \lambda \cdot P(Q|D) + (1 - \lambda) \cdot P(Q|G), \tag{7}$$

where, similar to Equation 2, $P(Q|D)$ is the probability that webpage D triggers query Q, and λ is the prior for the query being triggered from D. $P(Q|G)$ is the probability of the query being triggered by the background context. In our study, we will estimate $P(Q|G)$ using the user's search history H_u and general search history H_g for all the users:

$$P(Q|G) = \gamma P(Q|H_u) + (1 - \gamma)P(Q|H_g), \tag{8}$$

where $P(Q|H_u) \propto |Q \in H_u|$, and $P(Q|H_g) \propto |Q \in H_g|$ with $|Q \in H|$ denoting the frequency of query Q in H.

The model parameters can be estimated using the EM algorithm proposed in Section 4.1.3. After parameter estimation, we rank query candidates according to their probability $P(Q)$ as predictions/suggestions. Note that if we set $\lambda = 0$ and $\gamma = 0$, Equation 7 will only contain the $P(Q|H_g)$ part, which ranks queries based only on query frequency in the search logs. This special case is the same as the MPC model [1] used in query auto-completion task. If we only set $\lambda = 0$, the model becomes a personalized version of MPC, in which it jointly considers both search history from all users and the current user.

4.2.3 Features

To characterize how likely it is that D will trigger query Q for $P(Q|D)$, we use query-document matching and browsing-search history features as in previous work [7]. In addition, we design new features based on named entities, document position, and query freshness to the user, inspired by our in-depth analysis. All the proposed features are listed in Table 5.

Query-document matching features measure if the query is relevant to the document according to some pre-defined similarity measures. Here we check if the query appears in different parts of the document by *dMatch*, *hMatch*. We also measure word overlap between query and document, calculated as $\frac{|Q \cap D|}{|Q|}$, where Q, D are represented by the words after removing stopwords. In addition, we use features based on history browsing-search pairs. The intuition is that if many users issue the same query after browsing a certain webpage, then the webpage is very likely to trigger the particular query. Based on this, we use *qf* to count the frequency of a query being searched after users browse a webpage in the search history. Many popular navigational queries, such as "facebook", "google" are searched regardless of what webpages user have just browsed. In order to demote these non-triggered queries, we use *idf*, which represents the inverse frequency of different documents being browsed before users issue the query, in the same fashion as the conventional *idf* definition.

As we find named entities are important in our analysis, we use many entity-based features. We check if the query is or contains a named entity mentioned in the document by *eMatch* and *eContain*. We also use entity frequency to count the times of the query appearing as a named entity in the document by *eFreq* and *ehFreq*. Our analysis shows that users tend to search about 1) the beginning part of the a document and 2) new concepts that are fresh to the users. Therefore we use feature *pos* and *freshness* accordingly. Position in *pos* is normalized to 0 to 1 by word counts as described in Section 3.3.1, and set to 1 if the query does not appear in the news article.

Table 5: Pre-search context features for a webpage D triggering a search query Q.

Feature	Description
Query-document matching features	
dMatch	if Q appears in D
dOverlap	word overlap between Q and D
hMatch	if Q appears in the headline of D
hOverlap	word overlap between Q and the headline of D
Browsing-search history features	
qf	tf of Q in history browsing-search patterns
idf	idf of Q in history browsing-search patterns
qf.idf	$qf \times idf$
Newly designed features	
eMatch	if Q appears as a named entity in D
eContain	if Q contains a named entity mentioned in D
eOverlap	word overlap between Q and named entities in D
eFreq	entity frequency of Q in D
ehFreq	entity frequency of Q in the headline of D
pos	position of Q's first occurrence in D
freshness	if Q has been searched by the user before

5. EXPERIMENTS

In this section, we carry out a comprehensive set of experiments to evaluate our model on query prediction and query auto-completion. Particularly, we aim to empirically demonstrate that 1) pre-search context is useful for predicting user search intent and 2) our proposed model can effectively capture how queries are triggered with the hidden triggering relations in the real-world setting.

5.1 Experimental Settings

Dataset In order to evaluate our model in real-world scenarios, we continue to use the browsing and search logs described in Section 3.1. Here, we give a brief summary. We have two datasets: the first one is *history data*, which consists of 24 days of search and browsing logs. We only use it to extract user search history and click history. The second one is *experiment data*, which consists of 6,000 browsing-search

pairs. We use it to train and test models. Specifically, we split the experiment data set into three subsets: STUDY (S), TRAIN (T), and EVALUATE (E). We use TRAIN to train models and set aside STUDY and EVALUATE for testing.

Tasks We evaluate our model in two search tasks to demonstrate its effectiveness for search intent prediction: 1) *query prediction* aims to predict what a user is going to search (*i.e.*, her query) with the awareness of the pre-search context (*i.e.*, after browsing a webpage); 2) *query auto-completion* aims to suggest queries after a user browses a webpage and enters several prefix characters of a new query.

For both tasks, we use browsing-search pairs to evaluate. Specifically, for each browsing-search pair, we use the browsed webpage as the news pre-search context and the query issued after (within a limited time) as the correct answer. We emphasize that our research problem is predicting searches in this real-world setting, where the query may not actually be triggered by the browsed webpage – that is, it may be triggered by other uncaptured pre-search context such as locations and events.

To collect the candidate queries for both tasks, we pool the 100 most popular queries from the user's search history, the 100 most popular queries from search history across all users, and named entities appearing in the browsed webpage. To ensure there is an correct answer for comparing different models, we also include the following query in the candidate set. We believe this is a fair setting for comparison, which would reflect prediction effectiveness within the collected candidate set. The candidates for query auto-completion are almost the same as those for query prediction, except that now the query candidates are restricted to begin with the prefix characters.

Methods To demonstrate the usefulness of pre-search context and the effectiveness of our model, we compare against a wide range of baseline models as listed below.

- *Global Query Frequency* (GQF). This method estimates the probability $P(Q)$ that a query Q will be searched based on global search history, which is $P(Q|H_g)$ used in our model in Section 4.2.2. It is a widely used method [1] in query auto-completion. We use it as a baseline to compare the usefulness of the pre-search context and general search logs.
- *Global and User Query Frequency* (GUQF). This method estimates the probability $P(Q)$ that a user searches a query Q based on both global search history and user search history, which is $P(Q|G)$ used in our model in Section 4.2.2. We use it as a baseline to compare the usefulness of the pre-search context and user search history.
- *Leading Query Frequency* (LQF). This method estimates the probability $P(Q)$ that a user searches a query Q (with awareness of pre-search context) based on the frequency of Q in the leading queries of the browsed webpage in history. Particularly, $P(Q) \propto LQF(Q, D)$, where $LQF(Q, D)$ is the frequency of Q being a leading query for D in history. We use this method to demonstrate the difference between the triggering relationship and the clicking relationship, which is widely studied in the literature [8, 30].
- *Ranking SVM* (RSVM). Recent work [7] uses Ranking SVM to rank queries that are likely to be triggered by a browsed webpage as predictions for user searches. As mentioned before, in the real-world setting, searches may also be triggered by pre-search context other than browsed webpages. Therefore, using this model to blindly predict searches solely based on browsed webpage could be inadequate in the real-world setting. To test this hypothesis, we include this model as a baseline. In contrast to our model, this model requires expensive labeled triggered queries of each browsed webpages for training. In the experiments, we train the model with both the "real" triggered queries annotated by a human, denoted as RSVM-T, and pseudo triggered queries by assuming queries that appear in the browsed webpage are triggered by the webpage, denoted as RSVM-P. We use the same set of features as described in Table 5 for RSVM models.
- *Pre-search Context aware Intent Model* (PCIM). This is our proposed model described in Section 4.2.2. The model does not need labels for trigger/non-triggered queries for training, and aims to predict search intent in real-world settings where queries may be triggered by the browsed webpages *or* other pre-search context.

The parameters of all the models (*e.g.*, λ, γ and $\boldsymbol{w}$) are trained and tuned using the TRAIN dataset.

Metrics We use the following metrics to evaluate the effectiveness of different models.

- *Mean Reciprocal Rank (MRR)*. As both tasks aim to predict correct queries (only one for each evaluation case), we use the standard mean reciprocal rank [9] as our main measure.
- *Log-Likelihood*. In addition, we report how well a model "explains" the browsing-search pairs by the average of the log-likelihood of the queries in training and test data. For LQF, GQF and GUQF, when the query frequency is zero, we smooth the probability simply by assigning probability 1^{-10}. For RSVM, since Ranking SVM only produces scores $\boldsymbol{w}^T\boldsymbol{x}+b$ for ranking and does not provide probabilities, to calculate its log-likelihood we estimate its query probability based on the score as $P(Q) \propto \exp\{\boldsymbol{w}^T\boldsymbol{x}+b\}$.

For all the experiments, we also perform significance tests using paired t-test with 0.05 as the p-value threshold.

5.2 Experimental Results

5.2.1 Results for Query Prediction

We first evaluate different models for query prediction. Table 6 shows MRR of each method on STUDY and EVALUATE testing dataset. Table 7 shows the average log-likelihood of each model on both testing and training datasets (*i.e.*, STUDY, EVALUATE and TRAIN). Generally speaking, our model performs much better than all the baselines in terms of both MRR and log-likelihood on both test sets. The differences between our model and the baselines are statistically significant. We note that the log-likelihood of our model in training and testing data is close, which suggests that our model is not over-tuned. Next, we analyze the results in detail.

To begin with, we compare PCIM with the baselines that explore different data (*i.e.*, GQF, GUQF, and LQF). First, GQF performs worst in terms of MRR and the second worst in terms of log-likelihood, which clearly shows that the query frequency in general search history is not enough for query

Table 6: MRR for query prediction on STUDY(S) and EVALUATE(E).

	GQF	GUQF	LQF	RSVM-P	RSVM-T	PCIM
S	0.0426	0.1350	0.0498	0.0616	0.1206	**0.1705**
E	0.0429	0.1187	0.0447	0.0666	0.1242	**0.1556**

Table 7: Log-likelihood of different models on STUDY(S), EVALUATE(E) and TRAIN(T).

	GQF	GUQF	LQF	RSVM-P	RSVM-T	PCIM
S	-11.29	-10.62	-22.97	-4.95	-4.95	**-4.74**
E	-11.40	-10.83	-22.96	-4.94	-4.93	**-4.78**
T	-11.15	-10.49	-22.99	-4.95	-4.93	**-4.73**

prediction. Second, LQF performs worst in terms of log-likelihood and second worst in terms of MRR, which validates that clicking relationships between queries and webpages are different from the triggering relationships between webpages and queries and are not effective for query prediction. Third, GUQF largely improves over GQF and LQF in both metrics, which suggests that user personal search history provides valuable personal preference information for query prediction. PCIM, which captures the browsing pre-search context in additional to general and personal search history, performs the best and improves over the best baseline (GUQF) on MRR by 31% (on EVALUATE). It clearly demonstrates the effectiveness of pre-search context in query prediction.

Then, we compare PCIM with the state-of-the-art methods for triggered query prediction (*i.e.*, RSVM-P and RSVM-T). RSVM-P, trained on pseudo triggered queries, performs much worse than PCIM. Even though RSVM-T largely improves over RSVM-P due to the manually labeled triggered queries used in training, PCIM still largely improves over it on MRR by 25% (on Evaluate). These observations verify that RSVM models are inadequate or less effective than PCIM under the real-world setting where queries can be triggered from different types of pre-search context (*i.e.*, not only the browsed webpages) with the triggering source being hidden. The experiment results also demonstrate that PCIM can effectively learn from the unlabeled data to predict queries based on both the browsing pre-search context and background pre-search context that summarizes other un-captured pre-search context.

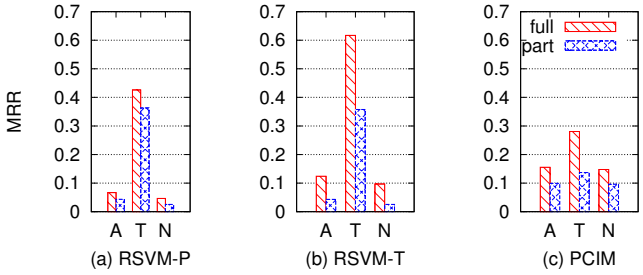
To justify our assumptions, we further break down the overall results by triggered (T) and non-triggered (N) cases (by manually labeling) in Table 8. We only report MRR on EVALUATE in these experiments because of space limitations, noting that MRR is consistant with other measures and the best aligned with our task. We have some interesting findings. First, GUQF largely improves GQF for non-triggered queries and has limited improvement for triggered queries, which shows the need for exploring pre-search context. This also validates our analysis in Section 3 that triggered queries are new and cannot be effectively predicted via exploring personal search history. Second, RSVM-P and RSVM-T perform better than PCIM for triggered queries but perform much worse than PCIM for non-triggered queries. The superior performance of RSVM models over PCIM on triggered queries implies that in the normally unrealistic setting of knowing searches are triggered from a specific pre-search context, supervised models that focus on learning queries triggered from a *single* type of pre-search context is a better choice than our unsupervised mixture model that tries to capture multiple types of pre-search context. However, we emphasize that the triggering relations are usually hidden in a real-world setting (Section 3.2). The system could not know if the current search will be triggered by the browsed webpage, and therefore could not apply RSVM models only on these triggered cases. The evaluation that combines both triggered and non-triggered queries in Table 6 and 7 reflects a realistic setting, in which we have shown that PCIM largely outperforms RSVM models.

Table 8: MRR for triggered(T) and non-triggered(N) queries.

	GQF	GUQF	LQF	RSVM-P	RSVM-T	PCIM
T	0.0090	0.0295	0.0699	0.425	**0.6169**	0.2807
N	0.0445	0.1228	0.0433	0.0463	0.0967	**0.1478**

Finally, we demonstrate the usefulness of our newly proposed features. Specifically, we compare the performance with and without the new features (eMatch, eContain, eOverlap, eFreq, ehFreq, pos, freshness, described in Table 5) on RSVM-P, RSVM-T and PCIM. The results are reported in Figure 2. We can see that for almost all the cases, excluding the proposed features results in large decreases in performance, clearly demonstrating the utility of our proposed features.

Figure 2: MRR for models with and without our proposed features. "full" and "part" stand for with and without proposed features respectively. "A", "T" and "N" stand for all cases, triggered cases and non-triggered cases, respectively.



5.2.2 *Results for Query Auto-Completion*

In this section, we report results of different models on the query auto-completion task. In Table 9, we report MRR of different models evaluated for different initial lengths (the number of characters that a user has typed in).

Table 9: MRR for query auto-completion on varying length of prefix characters.

	Len=1	Len=2	Len=3	Len=4	Len=5
RSVM-T	0.1129	0.1671	0.2318	0.2533	0.2618
GQF	0.0926	0.1647	0.2848	0.3640	0.4165
GUQF	0.1810	0.2441	0.3512	0.4213	0.4676
PCIM	**0.2675**	**0.3077**	**0.3797**	**0.4395**	**0.4779**

Generally speaking, the results confirm our findings from the previous experiment. GUQF improves over GQF, and PCIM further improves over GUQF and RSVM-T in query

auto-completion. Most importantly, the improvements of our model are consistent for any length of query prefix, and the differences are all statistically significant. The results again suggest that our model can effectively model pre-search context, and consistently outperforms baseline models. Further, we have some specific findings from this task. First, MRR for each model increases as prefix length increases, suggesting that the problem of query auto-completion become easier as the user types more letters. Second, the improvement for both GUQF over GQF and PCIM over GUQF decreases as initial length increases. This is because as a user types more and more letters, the additional information personal search history and pre-search context can provide becomes less and less helpful. However, from another point of view, this observation also suggests that pre-search context is particularly useful for the most challenging cases, in which the user has only typed one or two letters.

6. CONCLUSIONS

In this paper, we study the problem of predicting search intent based on pre-search context with hidden triggering relations, which advances the research of search intent prediction into a realistic setting. We make the following contributions to the problem.

- We provide a key insight that searches could be triggered by different types of pre-search context, and search systems often do not observe which particular one triggers each search (*i.e.*, hidden triggering relations).
- We conduct an in-depth analysis of how news pre-search context triggers searches based on real-world browsing and search logs. The analysis verifies the hidden nature of triggering relations between pre-search context and triggered queries, identifies interesting characteristics of them that lead to informative features, and provides guidelines for studying other pre-search context in the future.
- To address the challenge of hidden triggering relations, we developed an unsupervised generative mixture model to learn how queries are triggered by different types of pre-search context by taking advantage of large-size unlabeled data. We also identify a set of effective features which can be used in future work.
- Our experiment results show that: 1) pre-search context is useful in predicting search intent; 2) in the normally unrealistic case of knowing searches are triggered from a specific pre-search context, supervised models that focus on learning queries triggered from that specific pre-search context is more effective than our proposed unsupervised mixture model; but 3) in the realistic setting of hidden triggering relations, our proposed unsupervised model could learn effectively without expensive labels that are required in the supervised models, and predict search intents more accurately than them.

7. ACKNOWLEDGMENTS

This work was done during the first author's internship at Yahoo! Labs. It was also supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *WWW'11*.
[2] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short-and long-term behavior on search personalization. In *SIGIR'12*.
[3] I. Bordino, G. De Francisci Morales, I. Weber, and F. Bonchi. From machu_picchu to rafting the urubamba river: anticipating information needs via the entity-query graph. In *WSDM'13*.
[4] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *SIGIR'09*.
[5] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *WWW'09*.
[6] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *KDD'08*.
[7] Z. Cheng, B. Gao, and T.-Y. Liu. Actively predicting diverse search intent from user browsing behaviors. In *WWW'10*.
[8] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR'07*.
[9] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.
[10] S. Dumais, E. Cutrell, R. Sarin, and E. Horvitz. Implicit queries (iq) for contextualized search. In *SIGIR'04*.
[11] H. Feild and J. Allan. Task-aware query recommendation. In *SIGIR'13*.
[12] S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based user profiles for search and browsing. *Journal of Personalization Research, Special*, pages 1–3, 2002.
[13] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: user behavior as a predictor of a successful search. In *WSDM'10*.
[14] A. Hassan, Y. Song, and L.-w. He. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *CIKM'11*.
[15] K. Järvelin, S. L. Price, L. M. Delcambre, and M. L. Nielsen. Discounted cumulated gain based evaluation of multiple-query ir sessions. In *Advances in Information Retrieval*, pages 4–15. 2008.
[16] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. Evaluating multi-query sessions. In *SIGIR'11*.
[17] E. Kanoulas, M. M. Hall, P. Clough, B. Carterette, and M. Sanderson. Overview of the trec 2011 session track. In *TREC'11*.
[18] L. Li, Z. Yang, B. Wang, and M. Kitsuregawa. Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In *Advances in Data and Web Management*, pages 228–240. 2007.
[19] D. J. Liebling, P. N. Bennett, and R. W. White. Anticipatory search: using context to initiate search. In *SIGIR'12*.
[20] M. Rahurkar and S. Cucerzan. Predicting when browsing context is relevant to search. In *SIGIR'08*.
[21] M. A. Rahurkar and S. Cucerzan. Using the current browsing context to improve search relevance. In *CIKM'08*.
[22] K. Raman, P. N. Bennett, and K. Collins-Thompson. Toward whole-session relevance: exploring intrinsic diversity in web search. In *SIGIR'13*.
[23] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR'05*.
[24] Y. Song, X. Shi, R. White, and A. H. Awadallah. Context-aware web search abandonment prediction. In *SIGIR'14*.
[25] M. Speretta and S. Gauch. Personalized search based on user search histories. In *WI'05*.
[26] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW'04*.
[27] R. W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *SIGIR'09*.
[28] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM'10*.
[29] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *SIGIR'10*.
[30] J. Yi and F. Maghoul. Query clustering using click-through graph. In *WWW'09*.