Modelling time-aware search tasks for search personalisation

How to cite:

For guidance on citations see FAQs.

© 2015 The Authors
Version: Version of Record
Link(s) to article on publisher’s website:
http://dx.doi.org/doi:10.1145/2740908.2742714

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online’s data policy on reuse of materials please consult the policies page.
Modelling Time-aware Search Tasks for Search Personalisation

Thanh Vu, Alistair Willis, and Dawei Song∗
The Open University, Milton Keynes, UK
Tianjin University, Tianjin, P.R.C∗
firstname.surname@open.ac.uk

ABSTRACT
Recent research has shown that mining and modelling search
tasks helps improve the performance of search personali-
sation. Some approaches have been proposed to model a
search task using topics discussed in relevant documents,
where the topics are usually obtained from human-generated
online ontology such as Open Directory Project. A limita-
tion of these approaches is that many documents may not
contain the topics covered in the ontology. Moreover, the
previous studies largely ignored the dynamic nature of the
search task; with the change of time, the search intent and
user interests may also change.

This paper addresses these problems by modelling search
tasks with time-awareness using latent topics, which are au-
tomatically extracted from the task’s relevance documents
by an unsupervised topic modelling method (i.e., Latent
Dirichlet Allocation). In the experiments, we utilise the
time-aware search task to re-rank result list returned by
a commercial search engine and demonstrate a significant
improvement in the ranking quality.

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

Keywords: Time-aware Search Task; Search Personalisation;
Latent Topics

1. INTRODUCTION
Recently, search personalisation has been an active re-
search area and attracted increasing attention in literature
[1, 2, 4, 5]. Previous research has shown that mining and
modelling search task, which represents an atomic user in-
formation need, helps improve the performance of web
search personalisation [2, 5]. In the context of the web search
logs, a user may submit several queries within a search task
and handle several tasks within a search session [2].

A common approach is to model the search task with the
main topics discussed in the task’s relevant documents [2, 5].
In previous work, the topics of a document have often
obtained from a human-generated online ontology, such as
the Open Directory Project (ODP). One limitation of these
approaches is that many documents may not contain the
topics covered in the ontology. Furthermore, it needs ex-
ensive manual effort to determine the correct categories for
each document [1]. In addition, the dynamic nature of the
search task is largely ignored in previous search task mod-
elling studies (e.g., search task intent may be generalised or
specialised over the searching time).

To handle these problems, in this paper, we propose a uni-
fied framework to model search tasks which evolve from time
to time over a topic space (in Section 2.1). We utilise latent
topics automatically derived from the task’s relevant docu-
ments by an unsupervised topic modelling method (i.e., La-
et Dirichlet Allocation (LDA)) instead of using a human-generated
ontology [2, 5].

2. PERSONALISATION FRAMEWORK
2.1 Modelling Time-aware Search Tasks
Clustering Search Tasks in Sessions The primary
source of data for this study is a query logs from a com-
mercial search engine. A log sample consists of an anonymous
user identifier, a query, top-10 returned URLs, and clicked
results along with the user’s dwell time. To identify a ses-

To identify tasks from each search session, we apply the
Query Task Clustering approach (QTC) [2], which has a
desirable feature of extracting interleaved tasks within a
session. In general, the method works in two steps: first,
measure the similarities between query pairs to build an
undirected graph of queries within each search session; sec-
ond, cluster queries into tasks by dropping the weak edges
where the similarities are smaller than a threshold (0.9 in
our case). Specifically, each search task contains a set of
connected queries.

Extracting Latent Topics After getting search tasks
from the query logs, we first extract the relevant data of each
search task. We use the SAT criterion to identify satisfied
(SAT) clicks (as relevant data) from the query logs as either
a click with dwell time of at least 30 seconds or the last
click in a search session [1]. After that, we employ LDA
to extract latent topics (Z) from the relevant documents of
all search tasks. The LDA represents each document d as a
multinomial distribution over the topics Z.

Modelling a Time-aware Search Task We model a
search task with time-awareness as a multinomial distri-
bution over topic Z as follows. Formally, we denote the
search task set as $S$. Let $s$ denote an instance of $S$. Let $D_s = \{d_1, d_2, ..., d_n\}$ be a relevant document set of the search task $s$. We model the task $s$ (given $D_s$) as a distribution over the topic $Z$. Furthermore, since the search task intent may change over time, the more recent relevant documents could capture more about the search intent than the distant one. This characteristic can be modelled by introducing an exponential decay function $[1]$. In this paper, we model the probability of a topic $z$ given $s$ as a mixture of probabilities of $z$ given relevant document $d_i \in D_s$ as follows

$$p(z|s) = \frac{1}{K} \sum_{d_i \in D_s} \alpha^{t_{d_i}} \cdot p(z|d_i)$$

(1)

where $\alpha^{t_{d_i}}$ is the exponential decay function of $t_{d_i}$; $t_{d_i}$ is the time the searcher clicked on the document $d_i$ within $s$; $t_{d_i} = n$ indicates that $d_i$ is the $n^\text{th}$ recent relevant document; and $K = \sum_{d_i} \alpha^{t_{d_i}}$ is a normalisation factor.

2.2 Re-ranking Search Results

For each input query $q$, we utilise the time-aware search task $s$, to which the query belongs, to re-rank the first $n$ documents returned by a search engine. It is worth noting that $s$ captures the topical interests of the current user.

For each returned document $d$, we compute a similarity measure, $TaskScore$, between $d$ and $s$. Because both $d$ and $s$ are models as $X$ and $Y$ distributions over topic $Z$, respectively, we use Jensen-Shannon divergence to measure the similarity between the two distributions (i.e., $TaskScore = D_{JS}(X||Y)$). We consider the score as the personalised feature. We also extract other non-personalised features of $q$ and $d$. Table 1 describes the features.

### Table 1: Summary of the document features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskScore</td>
<td>The similarity score between the returned document and the search task</td>
</tr>
<tr>
<td>QueryNo</td>
<td>Total number of queries that have been submitted to the Search Engine</td>
</tr>
<tr>
<td>DocRank</td>
<td>Rank of the document on the original returned list</td>
</tr>
<tr>
<td>QuerySim</td>
<td>The cosine similarity score between the current query and the previous query</td>
</tr>
</tbody>
</table>

After extracting the document features, to re-rank the top $n$ returned documents, we employ a learning to rank algorithm (i.e., LambdaMART) to train ranking models as in [1][2].

3. EXPERIMENTS

**Dataset** We evaluate the approaches using the search results produced by a commercial search engine. The data used in our experiments is the query logs of 116 anonymous users in 15 days from 1st to 15th July 2012. We then partition the whole dataset into training and test sets. The training set contains the log data in the first 9 days and the test set contains the log data in the remaining days.

**Experimental Methodology** For evaluation, we use the SAT criterion to identify the satisfied clicks (SAT click) from the query logs. We assign a positive (relevant) label to a returned URL if it is a SAT click. Furthermore, similar to [1], we also assign a positive label to a URL if it is a SAT click in one of the repeated/modified queries in the same search session. The remaining of the top-10 URLs are assigned negative (irrelevant) labels. We use the rank positions of the positive labelled URLs as the ground truth to evaluate the search performance before and after re-ranking.

**Baselines** We name our proposed re-ranking model as TimeTask. Our first baseline, named as Default, is the original ranking of URLs returned by the search engine. We also construct a number of comparative baselines:

- **LongTerm, ShortTerm** are similar to our proposed re-ranking model. However, instead of modelling search tasks, these methods construct long-term and short-term user profiles respectively (see [2] for more detail).
- **StaticTask** is the non-temporal search task modelling method (that is, the decay parameter $\alpha = 1$).

**Overall Performance** We evaluate our proposed method by comparing the original rank list given by the commercial search engine and the re-ranked list given by our methods with four evaluation metrics: Mean Average Precision (MAP), Precision (P@k), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (nDCG@k). For each metric, the higher value indicates the better ranking. Table 2 shows that using time-aware search tasks, TimeTask achieves better performance than the original ranking (Default) as well as other strong baselines including non-temporal search task baseline (StaticTask). It also shows that ShortTerm (with time-awareness) gains advantage over StaticTask.

**Table 2: Overall performance of the methods. The improvements over Default are all statistically significant according to both paired t-test ($p < 0.01$) and Wilcoxon signed rank test ($p < 0.05$).**

<table>
<thead>
<tr>
<th>Models</th>
<th>Time-awareness</th>
<th>MAP</th>
<th>P@1</th>
<th>MRR</th>
<th>nDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td></td>
<td>0.6569</td>
<td>0.5414</td>
<td>0.6904</td>
<td>0.6854</td>
</tr>
<tr>
<td>LongTerm</td>
<td>✓</td>
<td>0.6685</td>
<td>0.5588</td>
<td>0.7057</td>
<td>0.7038</td>
</tr>
<tr>
<td>ShortTerm</td>
<td>✓</td>
<td>0.7217</td>
<td>0.6486</td>
<td>0.7677</td>
<td>0.7532</td>
</tr>
<tr>
<td>StaticTask</td>
<td></td>
<td>0.7140</td>
<td>0.6341</td>
<td>0.7581</td>
<td>0.7458</td>
</tr>
</tbody>
</table>

**TimeTask** ✓ 0.7504 0.6648 0.7775 0.7599

4. CONCLUSIONS

We have presented a method to model time-aware search tasks using latent topics. Each search task is represented as a distribution over the topics from which we extract the personalised feature and combine it with non-personalised features to learn a ranking function using LambdaMART.

We performed experiments on re-ranking search results returned by a commercial search engine. The results show that the ranking quality is improved significantly.

5. REFERENCES


