Temporal Latent Topic User Profiles for Search Personalisation

Thanh Vu[1], Alistair Willis[1], Son N. Tran[2], and Dawei Song[1,3]

[1] The Open University, Milton Keynes, United Kingdom
[2] City University London, London, United Kingdom
[3] Tianjin University, Tianjin, P.R.China
{thanh.vu,alistair.willis,dawei.song}@open.ac.uk,son.tran.1@city.ac.uk

Abstract. The performance of search personalisation largely depends on how to build *user profiles* effectively. Many approaches have been developed to build user profiles using topics discussed in relevant documents, where the topics are usually obtained from human-generated online ontology such as Open Directory Project. The limitation of these approaches is that many documents may not contain the topics covered in the ontology. Moreover, the human-generated topics require expensive manual effort to determine the correct categories for each document. This paper addresses these problems by using Latent Dirichlet Allocation for unsupervised extraction of the topics from documents. With the learned topics, we observe that the search intent and user interests are dynamic, i.e., they change from time to time. In order to evaluate the effectiveness of temporal aspects in personalisation, we apply three typical time scales for building *a long-term profile*, *a daily profile* and *a session profile*. In the experiments, we utilise the profiles to re-rank search results returned by a commercial web search engine. Our experimental results demonstrate that our temporal profiles can significantly improve the ranking quality. The results further show a promising effect of temporal features in correlation with click entropy and query position in a search session.

Keywords: User Profiles, Temporal Aspects, Latent Topics, Search Personalisation, Re-ranking.

1 Introduction

As one of the key components in advanced search engines (e.g., Google and Bing), *Search Personalisation* has attracted increasing attention [1,9,12,15,16,19]. The personalisation is expected to improve the usefulness of search algorithms. Unlike the search methods which don't use personalisation, personalised search engines utilise the personal data of each user to tailor search results, which depend not only on the input query but also on the user's interest (as context of the query). Such personal data can be used to construct a *user profile* which is crucial to effective personalisation.

Normally, one of the most common approaches is to represent the profile with the main topics discussed in documents which the user has previously clicked on [1,8,11,16,19]. The topics of a document are often obtained from a human-generated online ontology, such as the Open Directory Project (ODP) [1,11,19]. This approach has a limitation that many topics may not appear in the ontology. Moreover, it requires expensive manual effort to determine the correct categories for each document, as mentioned in [8]. In order to solve this problem, recent approaches [8,16] focus on learning latent topics from the relevant documents, using unsupervised models (i.e., Latent Dirichlet Allocation (LDA) [2]).

Latent topics have been successfully used to build user profiles, but little attention has been paid to the *temporal* aspects in the latent topic profiles, which reflect an important type of context. In this paper, we propose a novel temporal modelling approach for building user profiles from latent topics. We then carry out a comprehensive study on the effectiveness of temporal features in learning the topical interest of a user, with application to search results re-ranking. Our main goal is to address the following research questions: (1) Can temporal profiles help to improve search performance? and (2) How do temporal aspects affect the re-ranking quality?

To this end, we construct three temporal latent topic profiles for each user using the relevant documents with different time scales in the user's search history. We name the profiles as *session profile*, *daily profile* and *long-term profile*, as they are built from the topics learned from the documents within a session, a day and a whole history respectively. We note that the three profiles represent the user interest in different time scales (from short-term to long-term). In order to extract topics from the relevant documents, we employ the same approach proposed in [16] that utilises a topic modelling method (i.e., LDA [2]) to automatically derive *latent topics* instead of using a human-generated ontology as in [1,11,19].

The rest of this paper is structured as follows. In Section 2, we present the related work on user modelling for search personalisation. Section 3 describes our personalisation framework for building the temporal profiles and using the profiles to re-rank the returned result list. In Section 4, we describe our experiment setting. We then report the results in Section 5 and conclude the paper in Section 6.

2 Related Work

The user profile maintains the user's information on an individual level, typically based on the terms that represent user's search interests. To represent a user profile, Bennett *et al.* [1] mapped the user's interest onto a set of topics, which are extracted from large online ontologies of web sites, namely the ODP. This approach suffers from a limitation that many documents may not appear in the online categorisation scheme. Moreover, it requires expensive manual effort to determine the correct categories for each document. Harvey *et al.* [8] and Vu *et al.* [16] applied a latent topic model (i.e., LDA) to determine these topics. This means that the topic space is determined based purely on relevant documents extracted from query logs and does not require human involvement to define the topics. However, in their researches, the authors used all relevant documents extracted from the user's whole search history to construct the user profile (i.e., long-term profile). Moreover, they treated the relevant documents equally without considering temporal features (i.e., the time of documents being clicked and viewed).

The user interests could be long-term [6,8,14,16] or short-term [18,19]. Long-term interests, in the context of *IR* systems, are stable interests that can be exhibited for a long time in the user's search history. The long-term interests have been shown helpful for improving the search results [6,8,16]. Typically, the interests are represented as frequent terms or topics which have been extracted from the text of user's queries and clicked results. Alternatively, they can be also extracted from other personal data such as computer files and emails etc. [14]. In the application of re-ranking, [8,14,16], these terms/topics that represent long-term interests are used to re-rank relevant documents with the future queries.

Short-term interests, on the other hand, are temporary interests of a searcher during a relatively short time (e.g. in one or some continuous search sessions). The short-term interests are usually obtained from the submitted queries and the clicked documents in a search session and used to personalise the search within the session [18,19]. Bennett *et al.* [1] studied the interaction between long-term and short-term and found that the long-term behaviour provided advantages at the start of a search session while short-term session played a very important role in the extended search session. Furthermore, the combination of short-term and long-term interactions outperformed using either alone.

In this paper, in constrast to Bennett *et al.* [1] and White *et al.* [19], we apply LDA to automatically derive the latent topics from the user's relevant documents. Furthermore, in contrast to [8,16] as building a single user profile statically, we propose three temporal user profiles (i.e., long-term, daily and session profiles) which can represent both long-term and short-term user interests. It is worth noting that our long-term profile is different from Vu *et al.* [16] in term of considering the view-time of the relevant document (Section 3.2). We then thoroughly investigate the effectiveness of the proposed profiles in search personalisation.

3 Personalisation Framework

3.1 Extracting Topics from Relevant Documents

We briefly describe the method to extract topics from relevant documents, which was initially proposed in [16]. We first extract the relevant data of each user from the query logs. A log entry consists of an anonymous user-identifier, a submitted query, top-10 returned URLs, and clicked results along with the user's dwell time. We use the SAT criteria detailed in [7] to identify satisfied (SAT) clicks (as relevant data) from the query logs as either a click with a dwell time of at least 30 seconds or the last result click in a search session. To identify a *session*, we use the common approach of demarcating session boundaries by 30 minutes of user inactivity [11].

After that, we employ LDA [2] to extract latent topics (Z) from the SAT clicked documents (D) of all users. LDA represents each topic as a multinomial distribution over the entire vocabulary. Furthermore, each document is also described as a multinomial distribution over topics.

3.2 Constructing User Profiles

Modelling a User Profile Formally, the user variable is denoted as U. Let u denote an instance of U. We build a user profile based on the topics of the user's relevant documents. Let $D_u = \{d_1, d_2, .., d_n\}$ be a relevant document set of the user u. We define the user profile of u (given D_u) as a distribution over the topic Z. The probability of a topic z given u is defined as a mixture of probabilities of z given relevant document $d_i \in D_u$ as follows

$$p(z|u) = \sum\nolimits_{d_i \in D_u} \lambda_i p(z|d_i) \tag{1}$$

Here, $\sum_i \lambda_i = 1$ to guarantee that $\sum_z p(z|u) = 1$. The simple approach as used in Vu *et al.* [16] is to treat relevant documents equally when calculating $p(z|u)$. It means that $\lambda_1 = \lambda_2 = ... = \lambda_n = \frac{1}{|D_u|}$. Therefore, we have

$$p(z|u) = \frac{1}{|D_u|} \sum\nolimits_{d_i \in D_u} p(z|d_i) \tag{2}$$

Temporal weighting Since the search intent and user interest change over time, the more recent relevant documents could express more about the user interest than the distant one. This characteristic can be captured by introducing a decay function [18,1]. In this paper, instead of treating all the relevant documents equally (e.g. [16]), we model λ_i as the exponential decay function of t_{d_i}, which is the time the user u clicked on the document d_i, as follows

$$\lambda_i = \frac{1}{K} \alpha^{t_{d_i} - 1} \tag{3}$$

where $K = \sum_{d_i} \alpha^{t_{d_i} - 1}$ is a normalisation factor; $t_{d_i} = 1$ indicates that d_i is the most recent relevant (SAT click) document. By applying Eq. 3 to Eq. 1, we have

$$p(z|u) = \frac{1}{K} \sum\nolimits_{d_i \in D_u} \alpha^{t_{d_i} - 1} p(z|d_i) \tag{4}$$

Motivating example Previous work [8,16] on latent topic-based user profiles only used a single user profile (i.e., long-term profile). This work treated all the relevant documents equally and used the user's whole search history to construct the profile. In this paper, however, we treat the relevant documents temporally based on the viewing time of the user on the document. Furthermore, a single long-term profile cannot quickly represent the short-term interest of a user in a search session or in a specific day. For example, with a user having a strong law background, the long-term profile of the user has been constructed from thousands of law-related documents. On the first day of the World Cup (WC) 2014, even though she submitted WC-related queries and clicked on WC-related documents, the updated long-term profile cannot change promptly to express the football interest and does not seem to help personalising the WC-related queries. Therefore, apart from the long-term profile, we model two other profiles, namely *daily* and *session* profiles using the user's relevant documents in the current searching day and current search session respectively. It is worth clarifying that the long-term profile represents the permanent/long-term interest of the user. Otherwise, the session profile describes the provisional interest of the current user. The daily profile indicates the user interest over a searching day. Finally, we construct the three user profiles using different relevant datasets which change overtime as follows:

Long-term Profile We build the long-term user profile of u using relevant documents D_w extracted from the user's whole search history as follows

$$p_w(z|u) = \frac{1}{K}\sum\nolimits_{d_i \in D_w} \alpha^{t_{d_i}-1} p(z|d_i) \tag{5}$$

Daily Profile We build the daily user profile of u using relevant documents D_d extracted from the search history of u in the current day as follows

$$p_d(z|u) = \frac{1}{K}\sum\nolimits_{d_i \in D_d} \alpha^{t_{d_i}-1} p(z|d_i) \tag{6}$$

Session Profile We build the session user profile of u using relevant documents D_s extracted from the current search session of u as follows

$$p_s(z|u) = \frac{1}{K}\sum\nolimits_{d_i \in D_s} \alpha^{t_{d_i}-1} p(z|d_i) \tag{7}$$

3.3 Re-ranking Search Results using User Profiles

We utilise the user profiles to re-rank the original list of documents returned by a search engine. The detailed steps are as follows

(1) We download the top n ranked search results (as recorded in a data set of query logs) from the search engine for a query. We denote a downloaded web page as d and its rank in the search result list as $r(d)$.

(2) We then compute a similarity measure, $Sim(d|p)$, between each web page d and user profile p. Because both d and p are models as D, P distributions over topic Z, respectively, we use Jensen-Shannon divergence ($D_{JS}\lfloor.||.\rfloor$) to measure the similarity between the two probability distributions as follows

$$Sim(d|p) = D_{JS}\lfloor D||P\rfloor = \frac{1}{2}D_{KL}\lfloor D||M\rfloor + \frac{1}{2}D_{KL}\lfloor P||M\rfloor \tag{8}$$

Here $D_{KL}\lfloor.||.\rfloor$ is the Kullback-Leiber divergence and $M = \frac{1}{2}(D + P)$. After this step, we get three personalised scores, denoted as $f_w = Sim(d|p_w)$, $f_d = Sim(d|p_d)$, and $f_s = Sim(d|p_s)$, with respect to long-term, daily, and session profiles respectively. We consider the three scores as the personalised features of the document d.

(3) The personalised features only represent the user interest on a returned document. Therefore, apart from these features, we also extract other non-personalised features of input query q and the search result d. The full description of these features is presented in Table 1.

Table 1. Summary of the document features.

Feature	Description
Personalised Features	
LongTermScore	The similarity score between the document and the long-term profile
DailyScore	The similarity score between the document and the daily profile
SessionScore	The similarity score between the document and the session profile
Non-personalised Features	
DocRank	Rank of the document on the original returned list
QuerySim	The cosine similarity score between the current query and the previous query
QueryNo	Total number of queries that have been submitted to the Search Engine

(4) After extracting the document features, to re-rank the top n returned URLs instead of using a simple ranking function [16], we employ a learning to rank algorithm (LambdaMART [3]) to train ranking models. Among many learning to rank algorithms, LambdaMART has been regarded as one of the best performing algorithms [4], and has been chosen as the base learning algorithm in various state of the art approaches to search personalisation[4] [1,12,13,17]. However, it is worth noting that our proposed features are insensitive to ranking algorithm, thus any reasonable learning-to-rank algorithm would likely provide similar results.

4 Experimental Methodology

4.1 Dataset and Evaluation Methodology

Dataset In the experiment, 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 1166 anonymous users in four weeks, from 01^{st} July 2012 to 28^{th} July 2012. Each sample in the query logs consists of: an anonymous user identifier, an input query, the query time, top 10 returned URLs and clicked results along with the user's dwell time. We also download the content of these URLs for the learning of the topics.

We then partition the whole dataset into profiling, training and test sets. The profiling set is used to build the long-term user profile, the training set is for training the ranking model using LambdaMART and the test set is used for evaluation of the approaches. In particular, the profiling set contains the log

[4] Indeed, an ensemble of LambdaMART rankers won Track 1 of the 2010 Yahoo! *Learning to Rank Challenge* [5].

data in the first 13 days; the training set contains the query logs in next 2 days; and the test set contains the log data in the remaining 13 days. Table 2 shows the basic statistics on the three datasets.

Table 2. Basic statistics of the evaluation search log set.

Item	ALL	Profiling	Training	Test
#days	28	13	2	13
#queries	520010	240066	29834	236615
#distinct queries	176029	85641	12112	89445
#search session	94972	43462	5655	45886
#clicks	433277	200119	25805	207353
#SAT clicks	334227	154753	19513	159961
#SAT clicks/#queries	0.6427	0.6446	0.6541	0.6760

Evaluation Methodology For evaluation, we use the SAT criteria [7] 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[5]. The remainder 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. We also apply a simple pre-processing on these data sets as follows. At first, we remove the queries whose positive label set is empty from the dataset. After that we discard the domain-related queries (e.g. Facebook, Youtube). Finally, we normalise the relevance features (both personalised and non-personalised features) to zero mean and standard deviation (i.e., z-score) from the training set.

4.2 Experimental Settings

Personalisation Methods and Baselines We empirically investigate the effect of different temporal aspects in latent topic-based personalisation by using the three proposed profiles and their combination to generate the following features:

1. LongTermScore from long-term profile (LON)
2. DailyScore from daily profile (DAI)
3. SessionScore from session profile (SES)
4. AllScore from combination of three profiles (ALL)

We further combine these features with the non-personalised features to enrich the personalisation with relevant information from all users. As mentioned earlier, our first baseline, named as *Default*, is the search results (ranking of

[5] A query q' is a modification of query q if the returned URLs (top 10) of q' contains at least one SAT click of q.

URLs) returned by the commercial search engine, where we obtain the log data. The second baseline we would like to compare with is the combination of non-personalised features and the topic features proposed by Vu *et al.* [16], which does not take the temporal features into account. We named the second baseline as *Static*.

In the following we present the setting of LDA and LambdaMART for learning the topics and for learning the ranking function respectively. Note that in order to make a fair comparison we use the same topic distributions for all personalisation approaches and baselines.

LDA & LambdaMART We train the LDA model on the relevant documents extracted from the query logs, as mentioned in Section 3.1. The number of topics is decided by using a held-out validation set which consists of 10% of all the relevant documents. The selected number of topics is the one that gives the lowest perplexity value. We also use the validation set to select the temporal weighting parameter α.

The ranking function is learned using LambdaMART. After getting the features from the approaches, we randomly extract 10% of the training set for validation. We used the default setting for LambdaMART's prior parameters[6]. We follow the same model selection process as in [1,12].

Evaluation metrics The evaluation is based on the comparison between our personalised approaches and the baselines. For completeness, we use four evaluation metrics which are: Mean Average Precision (MAP), Precision (P@k), Mean Reciprocal Rank (MRR) and Normalized Discounted Cumulative Gain (nDCG@k). These are standard metrics which have been widely used for performance evaluation in document ranking [10]. For each evaluation metric, the higher value indicates the better ranking.

5 Experimental Results

5.1 Overall Performance

In this experiment, we analyse the effect of temporal aspects on latent topic profiles as proposed in Section 3 using six metrics: MAP, P@1, P@3, MMR, nDCG@5 and nDCG@10. Table 3 shows promising results when the temporal features are used to build user profiles. One can see that all three temporal profiles (i.e., session, daily, long-term profiles) have led to improvements over the original ranking and the use of non-temporal profile. Especially, the combination of all features (ALL) achieves the highest performance. This interesting result shows that a comprehensive user profile should capture different temporal aspects of the user's history. It should be noted that the improvements over the baselines reported in Table 3 are all significant with paired t-test of $p < 0.001$.

In the comparison between the temporal profiles, Table 3 shows that the session profile (SES) achieves better performance than the daily profile (DAI). It also shows that the daily profile (DAI) gains advantage over the long-term

[6] Specifically, number of leaves = 10, minimum documents per leaf = 200, number of trees = 100 and learning rate = 0.15.

Table 3. Overall performance of the methods.

Models	MAP	P@1	P@3	MMR	nDCG@5	nDCG@10
Default	*0.7494*	*0.6471*	*0.3320*	*0.7699*	*0.7805*	*0.8197*
Static	0.7460	0.6464	0.3289	0.7683	0.7751	0.8175
LON	0.7577	0.6601	0.3377	0.7813	0.7911	0.8267
DAI	0.7760	0.6897	0.3473	0.8016	0.8080	0.8406
SES	0.7936	0.7207	0.3537	0.8214	0.8238	0.8540
ALL	**0.7964**	**0.7283**	**0.3543**	**0.8254**	**0.8251**	**0.8563**

profile (LON). This indicates that the short-term profiles capture more details of user interest than the longer ones. The results are also consistent with what has been found in [1]. The difference is that our profiles are based on the learned latent topics while they use the ODP.

5.2 Click Entropies

In search personalisation, click entropy plays an important role in deciding the search performance. In [6], Dou et al. have argued that a small click entropy may deteriorate the quality of the search results. The click entropy of a query is defined as:

$$ClickEntropy(q) = \sum\nolimits_{d \in D_q} -p(d|q)\log_2 p(d|q) \qquad (9)$$

Here D_q is a collection of web pages which are clicked for the distinct query q, and $p(d|q)$ is the percentage of the clicks on document d among all the clicks for q. A smaller query click entropy value indicates more agreement between users on clicking a small number of web pages. In this paper, we are also interested in investigating the effect of the click entropy on the performance of the temporal latent topic profiles. In the experimental data, about 67.25% and 16.34% queries have a low click entropy from 0 to 0.5 and from 0.5 to 1 respectively; 10.05% and 3.95% queries have a click entropy from 1 to 1.5 and from 1.5 to 2 respectively; and only 2.41% queries have a high click entropy (≥ 2).

In Figure 1, we show the improvement of the temporal profiles over the *Default* ranking from the search engine in term of MAP metric for different magnitudes of click entropy. Here the statistical significance is also guaranteed with the use of paired t-test ($p < 0.001$). The results show that when users have more agreement over clicked documents, with respect to smaller value of click entropy, the re-ranking performance is only slightly improved. For example, with click entropy between 0 and 0.5, the improvement of the MAP metric from long-term profile is of only 0.39%, in comparison with the original search engine. One may see that the effectiveness of the temporal profiles is increasing proportionally according to the value of click entropy. In particular, the improvement of personalised search performance increases significantly when the click entropy becomes larger, especially with click entropies ≥ 0.5, and the highest improvements are achieved when click entropies are ≥ 2. This result contributes a case study on temporal latent topic profiles to the study of click entropy for personalisation besides the static latent topic profile [16].

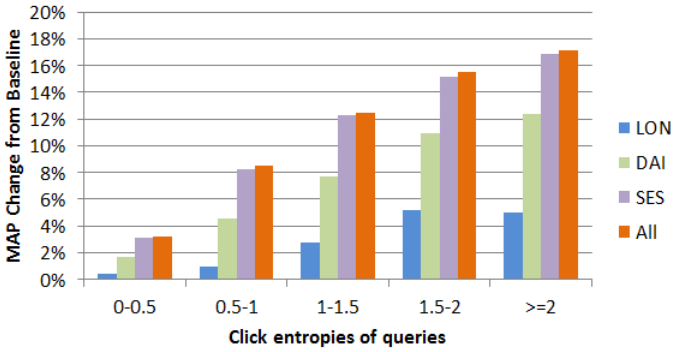

Fig. 1. Search performance improvements over *Default* with different click entropies.

5.3 Query Positions

A query usually has a broader influence in a search session than only returning a list of URLs. The position of a query in a search session is also important because it may be fine-tuned by a user after the unsatisfactory results from previous queries. Therefore, in order to get into the insights of the user's information need, a search engine should take into account the position of an input query in a search session. In this experiment we aim to study whether the position of a query has any effect on the performance of the temporal latent topic profiles. For each session, we label the queries by their positions during the search. The first five queries are numbered from one to five according to the order of the time that they have been entered to the search engine, the remaining queries are labelled as ≥ 6, similarly as in [1].

We show the MAP performances of the temporal latent topic profiles for different query positions in Figure 2. From the MAP values, we can see that the first query always received higher satisfaction than the others. It shows that the advanced search engine where we extracted the logs has managed to produce reasonably relevant results at the first query. The higher query positions achieve smaller value of MAP in a search session, which can be explained as users tend to search for supplementary information after the first query, and that the latter queries are so similar to the previous one that the search results contains many URLs which have already appeared in the previous search result. Our result is consistent to what has been mentioned in [19].

Note that we cannot build a session profile for the first query because there is no previously observed relevant document for the query. For long-term and daily profiles, we found that their search performances are similar to the search engine performance of the first query. This can be explained by the fact that the single long-term and daily profiles are diverse and cannot sufficiently represent the user recent interests for the first query. Furthermore, as shown in Figure 2, the search engine satisfies most the user's information need for the first query (MAP value of 0.8353 out of 1). However, for the next queries in the search

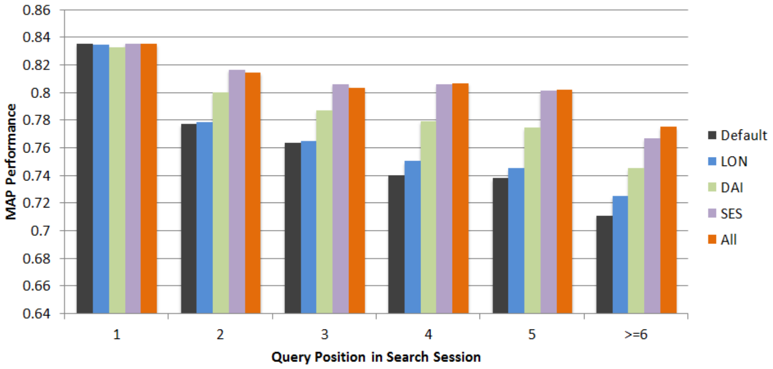

Fig. 2. Performances of the methods by position of query in search session.

session, the temporal latent topic profiles show a significant improvement. It shows that temporal profiles can quickly adapt to represent the user interest. For example, the session profile achieves the highest performance on the second and the third queries in a session whilst the combination of profiles outperforms the other models on the queries from the fourth positions. This new result is interesting because it shows that the temporal features can help tuning the search performance in further queries which has not been done successfully by the original search engine.

6 Conclusions

We have presented a study on the temporal aspects for building user profiles with latent topics learned from the documents. For each user, we used relevant documents at different time scales to build long-term, daily, and session profiles. Each user profile is represented as a distribution over latent topics from which we extract the features and combine them with non-personalised features to learn a ranking function using LambdaMART. We performed a set of experiments to study the effectiveness of the temporal latent topic-based profiles.

The results showed that the temporal features help improve search performance over the competitive ranker of the original search engine and over the static latent topic profile. We also found that the session profile captures the most interests of a user and is able to generate helpful features for learning the re-ranking function. The best performance was achieved by the combination of all three temporal profiles, indicating that a good personalisation should take into account all temporal aspects from user's search history. Other experimental results confirmed that the impact of the query's click entropy on temporal latent topic profile is similar to that on the static latent topic profile. Finally, another interesting finding is the usefulness of the temporal profile in tuning the search results for the next queries in a search session.

Acknowledgements This work is partially funded by the Chinese National Program on Key Basic Research Projects (973 Program, grant no. 2013CB329304, 2014CB744604) and the National Science Foundation of Chinese (grant no. 61272265). We thank the reviewers for their valuable comments. We would also like to thank Jingfei Li for providing the data & his help in pre-processing it.

References

1. 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*, pages 185–194. ACM, 2012.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, pages 993–1022, 2003.
3. C. J. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS*, pages 193–200. MIT Press, 2007.
4. C. J. C. Burges. From ranknet to lambdarank to lambdamart : An overview. Technical Report MSR-TR-2010-82, Microsoft Research, July 2010.
5. O. Chapelle, Y. Chang, and T. Liu. Yahoo! learning to rank challenge overview. In *JMLR*, pages 1–24, 2011.
6. Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW*, pages 581–590. ACM, 2007.
7. S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, pages 147–168, 2005.
8. M. Harvey, F. Crestani, and M. J. Carman. Building user profiles from topic models for personalised search. In *CIKM*, pages 2309–2314. ACM, 2013.
9. A. Hassan and R. W. White. Personalized models of search satisfaction. In *CIKM*, pages 2009–2018. ACM, 2013.
10. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.
11. K. Raman, P. N. Bennett, and K. Collins-Thompson. Toward whole-session relevance: Exploring intrinsic diversity in web search. In *SIGIR*, pages 463–472, 2013.
12. M. Shokouhi, R. W. White, P. Bennett, and F. Radlinski. Fighting search engine amnesia: Reranking repeated results. In *SIGIR*, pages 273–282. ACM, 2013.
13. Y. Song, X. Shi, R. White, and A. H. Awadallah. Context-aware web search abandonment prediction. In *SIGIR*, pages 93–102. ACM, 2014.
14. J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR*, pages 449–456. ACM, 2005.
15. J. Teevan, M. R. Morris, and S. Bush. Discovering and using groups to improve personalized search. In *WSDM*, pages 15–24. ACM, 2009.
16. T. T. Vu, D. Song, A. Willis, S. N. Tran, and J. Li. Improving search personalisation with dynamic group formation. In *SIGIR*, pages 951–954. ACM, 2014.
17. H. Wang, Y. Song, M.-W. Chang, X. He, A. Hassan, and R. W. White. Modeling action-level satisfaction for search task satisfaction prediction. In *SIGIR*, pages 123–132. ACM, 2014.
18. R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM*, pages 1009–1018. ACM, 2010.
19. R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing Personalized Search by Mining and Modeling Task Behavior. In *WWW*, pages 1411–1420. ACM, 2013.