Bridging Topic Modeling and Personalized Search

Wei Song Yu Zhang Ting Liu Sheng Li
School of Computer Science
Harbin Institute of Technology
{wsong, yzhang, tliu, lisheng}@ir.hit.edu.cn

Abstract

This work presents a study to bridge topic modeling and personalized search. A probabilistic topic model is used to extract topics from user search history. These topics can be seen as a roughly summary of user preferences and further treated as feedback within the KL-Divergence retrieval model to estimate a more accurate query model. The topics more relevant to current query contribute more in updating the query model which helps to distinguish between relevant and irrelevant parts and filter out noise in user search history. We designed task oriented user study and the results show that: (1) The extracted topics can be used to cluster queries according to topics. (2) The proposed approach improves ranking quality consistently for queries matching user past interests and is robust for queries not matching past interests.

1 Introduction

The majority of queries submitted to search engines are short and ambiguous and the users of search engines often have different search intents even when they submit the same query (Janse and Saracevic, 2000)(Silverstein and Moricz, 1999). The "one size fits all" approach fails to optimize each individual's specific information need. Personalized search has be viewed as a promising direction to solve the "data overload" problem, and aims to provide different search results according to the specific preference of an individual(Pitkow and Breuel, 2002). Information retrieval (IR) communities have developed models for context sensitive search and related applications (Shen and Zhai, 2005a)(White and Chen, 2009).

The search context includes a broad range of information types such as a user's background, his personal desktop index, browser history and even the context information of a group of similar users (Teevan, 2009). In this paper, we exploit the user search history of an individual which contains the past submitted queries, results returned and the click through information. As described in (Tan and Zhai, 2006), search history is one of the most important forms of search context. When dealing with search history, distinguishing between relevant and irrelevant parts is important. The search history may contain a lot of noisy information which can harm the performance of personalization (Dou and Wen, 2007). Hence, we need to sort out relevant and irrelevant parts to optimize search personalization.

In this paper, we propose a topic model based approach to study users' preferences. The main contribution of this work is modeling user search history with topics for personalized search. Our approach mainly consists of two steps: topic extraction and relevance feedback. We assume that a user's search history is governed by the underlying hidden properties and apply probabilistic Latent Semantic Indexing (pLSI) (Hofmann, 1999) to extract topics from user search history. Each topic indexes a unigram language model. We model these extracted topics as feedback in the KL-Divergence retrieval framework. The task is to estimate a more accurate query model based on the evidence from user feedback. We distin-

Coling 2008: Poster Volume, pages 1167–1175,
Beijing, August 2010

guish relevant parts from irrelevant parts in search history by focusing on the relevance between topics and query. The closer a topic is to the current query, the more it contributes in updating the query model, which in turn is used to rerank the documents in results set.

2 Related Work

2.1 Personalized IR

Personalized search is an active ongoing research direction. Based on different representations of user profile, we classify approaches as follows:

Taxonomy based methods: this approach maps user interests to an existing taxonomy. ODP[1] is widely used for this purpose. For example, by exploiting the user search history, (Speretta and Gauch, 2005) modeled user interest as a weighted concept hierarchy created from the top 3 level of ODP. (Havelivala, 2002) proposed the "topic sensitive pagerank" algorithm by calculating a set of PageRanks for each web page on the top 16 ODP categories. (Qiu and Cho, 2006) further improved this approach by building user models from user click history. In recent studies, (Xu S. and Yu, 2008) used ODP categories for exploring folksonomy for personalized search. (Dou and Wen, 2007) proposed a method that represent user profile as a weighting vector of 67 predefined topic categories provided by KDD Cup-2005. Taxonomy based methods rely on a predefined taxonomy and may suffer from the granularity problem.

Content based methods: this category of methods use traditional text presentation model such as vector space model and language model to express user preference. Rich content information such as user search history, browser history and indexes of desktop documents are explored. The user profiles are built in the forms of term vectors or term probability distributions. For example, (Sugiyama and M., 2004) represented user profiles as vectors of distinct terms and accumulated past preferences. (Teevan and Horvitz, 2005) constructed a rich user model based on both search-related information, such as previously issued queries, and other information such as documents and emails a user had read and created. (Shen and Zhai, 2005b) used browsing histories and query sessions to construct short term individual models for personalized search.

Learning to rank methods: (Eugene and Susan, 2005) and (Eugene and Zheng, 2006) incorporated user feedback into the ranking process in a *learning to rank* framework. They leveraged millions of past user interaction with web search engine to construct implicit feedback features. However, this approach aims to satisfy majority of users rather than individuals.

2.2 Probabilistic Topic Models

Probabilistic topic models have become popular tools for unsupervised analysis of document collection. Topic models are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over words (Steyvers and Griffiths, 2007). These topics are interpretable to a certain degree. In fact, one of the most important applications of topic models is to find out semantic lexicons from a corpus. One of the most popular topic models, the probabilistic Latent Semantic Indexing Model (pLSI), was introduced by Hofmann (Hofmann, 1999) and quickly gained acceptance in a number of text modeling applications. In this study, pLSI is used to discover the underlying topics in user search history. Though pLSI is argued that it is not a complete generative model, we used it because it does not need to generate unseen documents in our case and the model is much easier to be estimated compared with sophisticated models such as LDA(David M. Blei and Jordan, 2003).

2.3 Model based Relevance Feedback

Our work is also related to language model based (pseudo) relevance feedback (Zhai and Lafferty, 2001b) and shares the similar idea with (Tan B. and Zhai, 2007). The differences are: (1) The feedback source is user search history rather than top ranked documents for a query. (2) We make use of user implicit feedback rather than explicit feedback. (3) The topics in search history could be extracted offline and updated periodically. Additionally, these topics provide an informative picture of user search history.

[1] Open Directory Project, http://dmoz.org/

Table 1: An illustration of topics extracted from a user's search history. Terms with highest probabilities are listed below each topic.

Topic 2	Topic 3	Topic 9	Topic 16
climb	movie	swim	cup
0.032	0.091	0.044	0.027
setup	download	ticket	world
0.022	0.078	0.032	0.022
equipment	dvd	notice	team
0.020	0.061	0.019	0.016
practice	watch	travel	brazil
0.009	0.060	0.016	0.011
player	cinema	hotel	storm
0.006	0.038	0.008	0.007

3 Proposed Approach

3.1 Main Idea

A user's search history usually covers multiple topics. It is crucial to distinguish between relevant and irrelevant parts for optimizing personalization. We propose a topic model based method to achieve that goal. First, we construct a document collection revealing user intents according to the user's past activities. A probabilistic topic model is applied on this collection to extract latent topics. Then the extracted topics are used as feedback. The query model is updated by highlighting the topics highly relevant to current query. Finally, the search results are reranked according to the relevance to the updated query model. Table 1 shows 4 topics extracted from a user's search history. Each topic is a unigram language model. The terms with higher probabilities belonging to each topic are listed. We can predict that the user has interests in both *movie* and *football*. However, when the user submits a query about *world cup*, the topic 16 is given higher preference for estimating a more accurate query model.

3.2 Topic Extraction from Search History

Individual's search history consists of all the past query units. Each query unit includes query text, returned search results (with title, snippets and URLs) and click through information. Here, we concatenated the title and snippet of each search result to form a document being considered as a whole. The whole search history can be seen as a collection of documents. Obviously, many documents in the collection may fail to satisfy the user's information need and are uncertain for discovering the user's preferences. Therefore, the first task is to select proper documents in search history as the preference collection for topic discovery.

3.2.1 Preference Collection

An intuitive solution is to use the documents that are clicked by the user. The assumption is that a user clicks on a result only if he is interested in the document. However, user click is sparse in real search environments and the documents not clicked by the user may also be relevant to the user's information need. We assumed that the user had only one search intent for a submitted query. To enhance this coherence within a query unit, we created only one super-document for a query unit as follows: if a query unit had clicked documents, then we concatenated these document to form a preferred document. Otherwise, we selected the top *n* documents from the search results and concatenated them as a preferred document. That is motivated by the idea of pseudo relevance feedback (Lavrenko and Croft, 2001) and used here for alleviating data sparsity. Pseudo relevance feedback is sensitive to the number of feedback documents. In this work, *n* is set to 3, because the average clicks for a query is not more than 3. By this way, we got a preference collection whose size is the same as the number of past queries.

3.2.2 Topic Extraction

Given the collection of preferred documents, we applied pLSI on this collection to extract underlying topics. We define the collection as $C=\{d_1,d_2,\ldots,d_M\}$, where d_i corresponds to the ith query unit, and M is the size of the collection. Each query unit is viewed as a mixture of different topics. It is reasonable in reality. For example, a news document about "*play basketball with obama*" might be seen as a mixture of topics "*politics*" and "*sports*".

Modeling: The basic idea of pLSI is to treat the words in each document as being generated from a mixture model where the component models are topic word distributions. Let k be the num-

ber of topics which is assumed known and fixed. θ_j is the word distribution for topic *j*. We extract topics from collection *C* using a simple probabilistic mixture model as described in (Zhai and Yu, 2004). A word *w* within document *d* can be viewed as generated from a mixture model:

$$p_d(w) = \lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j) \quad (1)$$

where θ_B is the background model for all the documents. The background model is used to draw common words across all the documents and lead to more discriminative and informative topic models, since θ_B gives high weights to non-topical words. λ_B is the probability that a term is generated from the background model which is set to be a constant. To draw more discriminative topic models, we set λ_B to 0.95. Parameter $\pi_{d,j}$ indicates the probability that topic *j* is assigned to the specific document *d*, where $\sum_{j=1}^{k}\pi_{d,j}$=1.

Parameter estimation: The parameters we have to estimate including the background model θ_B, $\{\theta_j\}$ and $\{\pi_{d,j}\}$. θ_B is maximum likelihood estimated (MLE) using all available text in our data set so that it is a fixed distribution. The other parameters to be estimated are $\{\theta_j\}$ and $\{\pi_{d,j}\}$. The log-likelihood of document *d* is:

$$\log p(d) = \sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)] \quad (2)$$

The log-likelihood of the whole collection *C* is:

$$\log(C) = \sum_{d\in C}\sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)] \quad (3)$$

The Expectation-Maximization (EM) algorithm (Dempster and Rubin, 1977) is used to find a group of parameters maximizing equation (3). The updating formulas are:

E-Step:

$$p(z_{d,w}=B) = \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)}$$

$$p(z_{d,w}=j) = \frac{\pi_{d,j}^{(m)} p^{(m)}(w|\theta_j)}{\sum_{j'=1}^{k}\pi_{d,j'}^{(m)} p^{(m)}(w|\theta_{j'})}$$

M-Step:

$$\pi_{d,j}^{(m+1)} = \frac{\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j)}{\sum_{j'=1}^{k}\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j')}$$

$$p^{(m+1)}(w|\theta_j) = \frac{\sum_{d\in C} c(w,d)(1-p(z_{d,w}=B))\,p(z_{d,w}=j)}{\sum_{d\in C}\sum_{w'\in V} c(w',d)(1-p(z_{d,w'}=B))\,p(z_{d,w'}=j)}$$

where $c(w,d)$ denotes the number of times *w* occurs in *d*. A hidden variable $z_{d,w}$ is introduced for the identity of each word. $p(z_{d,w}=B)$ is the probability that the word *w* in document *d* is generated by the background model. $p(z_{d,w}=j)$ denotes the probability that the word *w* in document *d* is generated using topic *j* given that *w* is not generated from the background model. Informally, the EM algorithm starts with randomly assigning values to the parameters to be estimated and then alternates between E-Step and M-Step iteratively until it yields a local maximum of the log likelihood.

Interpretation: As shown in equation (1), a word can be viewed as a mixture of topics. From the updating formulas, we can see that the dominant topic of a word depends on both itself and the context. The word tends to have the same topic with the document containing it. While the probability of assigning topic *j* to document *d* is estimated by aggregating all the fractions of words generated by topic *j* in document *d*. We can explain it in a more intuitive way with in our application. As we know, the queries are usually ambiguous. A classic example is *"apple"* which may refer to a kind of fruit, apple Inc, apple electric products, etc. Therefore, it is reasonable to assume that each word belongs to multiple latent semantic properties. If a returned result contains *"apple"* and other words like *"computer"*, *"ipod"* , etc. The word *"apple"* in this result tends to have the same topic distributions with *"computer"* and

'ipod". If the user clicks the result, we can predict that the user's real preference about query *"apple"* is related to electric products having a high probability. Further, if *"apple"* occurs frequently in many documents related to electric products, it obtains a higher probability in this topic. As a result, we not only know user's interest in electric products, but also find a preference to *"apple"* brand.

Since a document's topic depends on the words it contains, two documents with similar word distributions have similar topic distributions. In other words, each topic is like a bridge connecting queries with similar intents. In summary, the topic extraction process plays a role in our application for finding user preference, highlighting discriminative words and connecting queries with similar intents.

3.3 Topics as Feedback

The topics extracted from search history are considered as a kind of feedback. Since topic models actually are extensions of language models, we use such feedback within the KL-Divergence retrieval model (Xu and Croft, 1999)(Zhai and Lafferty, 2001b) that is a principled framework to model feedback in the language modeling approach. In this framework, feedback is treated as updating the query language model based on extra evidence obtained from the feedback sources. The information retrieval task is to rank documents according to the KL divergence $D(\theta_q||\theta_d)$ between a query language model θ_q and a document language model θ_d. The KL divergence is defined as:

$$D(\theta_q||\theta_d) = \sum_{w \in V} p(w|\theta_q) \log \frac{p(w|\theta_q)}{p(w|\theta_d)} \quad (4)$$

where *V* denotes the vocabulary. We estimate the document model θ_d using Dirichlet estimation (Zhai and Lafferty, 2001a):

$$p(w|\theta_d) = \frac{c(w,d) + \mu p(w|\theta_C)}{|d| + \mu} \quad (5)$$

where $|d|$ is document length, $p(w|\theta_C)$ is collection language model which is estimated using the whole data collection. μ is the Dirichlet prior that is set to 20 in this work. The updated query model is defined as:

$$\begin{aligned} p(w|\theta_q) &= \lambda p_{ml}(w|\theta_q) \\ &+ (1-\lambda) \sum_{j=1}^{k} p(w|\theta_j) p(z=j|q) \end{aligned} \quad (6)$$

where $p_{ml}(w|\theta_q)$ is the MLE query model. $\{\theta_j\}$ represents a set of extracted topics each of which is a unigram language model. λ is used to balance the two components. z is a hidden variable over topics. The task is to estimate the multinomial topic distribution $p(z|q)$ for query *q*. Since pLSI does not properly provide a prior, we estimate $p(z=j|q)$ as:

$$\begin{aligned} p(z=j|q) &= \frac{p(q, z=j)}{\sum_{j'=1}^{k} p(q, z=j')} \\ &\propto \frac{sim(\theta_q, \theta_j)}{\sum_{j'=1}^{k} sim(\theta_q, \theta_{j'})} \end{aligned} \quad (7)$$

Since the query text is usually very short, it is not easy to make a decision based on query text alone. Instead, we concatenate all the available documents in returned result set to form a superdocument. A language model is estimated for it. We convert both the document language model and topic models into weighted term vectors and use cosine similarity as the *sim* function. $p(z|q)$ plays an import role here as it determines the contribution of topics. The topics with higher similarity with current query contributes more in updating query model. This scheme helps to filter out noisy information in search history.

4 Evaluation and Discussion

4.1 Data Collection

To the best of our knowledge, there is no public collection with enough content information and user implicit feedback. We decided to carry out a data collection. Due to the difficulty to describe and evaluate user interests implicitly, we predefined some user interests and implemented a search system to collect user interactions.

The predefined interests belong to 5 big categories namely *Entertainment*, *Computer* & *Internet*, *Sports*, *Health* and *Social life*. Each interest is a kind of user preference such as "movies"

Table 2: An example of predefined user interests and tasks

category	Enterntainment
interest	movies
task1	search for a brief introduction of your favorite movie
task2	search for an introduction of an actor or actress you like
task3	search for movies about "artificial intelligence"

Table 3: Statistics of the data collection

user	1	2	3	4	5
#queries	218	256	177	206	311
#big category	5	5	5	5	5
#interest	25	25	25	25	25
#tasks	100	100	100	100	100
avg.#relevant results	4.17	4.22	3.89	4.12	3.24
avg.#clicked results	2.37	2.21	2.71	1.98	2.42

and "outdoor sports". For each interest, we designed several tasks each of which had a goal. Table 2 illustrates an example of a predefined user interest and related tasks. The volunteers were asked to find out the information need according to the tasks. Though we defined these interests and tasks, we did not impose any constraint on the queries. The volunteers could choose and reformulate any query they thought good for finding the desired information. But we did try to increase the possibility that a user might issue ambiguous queries by designing tasks like "search for movies about artificial intelligence" which was categorized to interest "movies", but also related to computer science.

To collect the user interaction with search engine, we implemented a Lucene based search system on Tianwang terabyte corpus(Yan and Peng, 2005). Five volunteers were asked to submit queries to this system to find information satisfying the tasks of each interest. The system recorded users' activities including submitted queries, returned search results (with title, snippet and URL) and users' click through information. When the user finished a task, he clicked a button to tell the system termination of the session containing all the queries and activities related to this task. After finishing all the tasks, the volunteers were asked to judge the top 20 results' relevance (relevant or not relevant) for each query according to the search target. Each volunteer submitted 233 queries on average. Table 3 presents some statistics of this collection.

4.2 Evaluating Topic Extraction

It is not easy to assess the quality of topics, because topic extraction is an unsupervised process and difficult to give a standard answer. Therefore, we view the topic extraction as a clustering problem that is to organize queries into clusters. To group queries into clusters through extracted topics, we use $\hat{j} = \arg\max_j \pi_{d,j}$ to assign a query to the $\hat{j}$th topic. Each topic corresponds to a cluster. All the queries are divided into *k* clusters. Based on the data collection, we setup the golden answers according to the predefined interests. We view all the queries belonging to a predefined interest(which includes multiple tasks) form a cluster which helps us to build a golden answer with 25 clusters in tatal.

One purpose of making use of topics in search history is to find more relevant parts and reduce the noise. We hope that the extracted topics are coherent. That is, a cluster should contain as many queries as possible belonging to a single interest. To evaluate coherence, we adopt *purity* (Zhao and Karypis, 2001), a commonly used metric for evaluating clustering. The higher the purity is, the better the system performs. We compare our method (denoted as PLSI) against the k-means algorithm(denoted as K-Means) on the preference collection.

Figure 1 shows the overall purity with different number of topics. Our method gained better performance than k-means algorithm consistently. It is effective to discover and organize user interests. Besides, as illustrated in Table 1, our method is able to give higher probability to discriminative words of each topic that provides a clear picture of user search history. This leads to an emergence of novel approaches for personalized browsing.

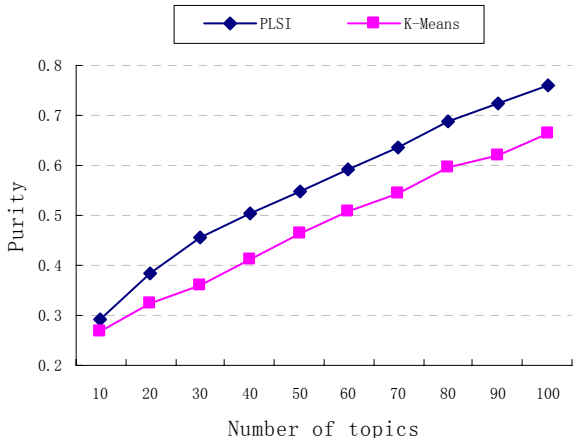

Figure 1: Average purity over 5 users gained by both PLSI and K-Means with different number of topics(clusters).

4.3 Evaluating Result Reranking

4.3.1 Metric

To quantify the ranking quality, the Discounted Cumulative Gain (DCG) (Jarvelin and Kekakainen, 2000) is used. DCG is a metric that gives higher weights to highly ranked documents and incorporates different relevance levels by giving them different gain values.

$$DCG(i) = \begin{cases} G(1), & \text{if } i = 1 \\ DCG(i-1) + \frac{G(i)}{\log(i)}, & \text{otherwise} \end{cases}$$

In our work, we use $G(i) = 1$ for the results labeled as relevant by a user and $G(i) = 0$ for the results that are not relevant. The average normalized DCG (NDCG)over all the test queries is selected to show the performance.

4.3.2 Systems

We evaluated the performance of following systems:

PLSI: The proposed method. The history model was a weighted interpolation over topics extracted from the preference collection described in session 3.2.1.

PSEUDO: From each query unit, we selected top n documents as pseudo feedback. The language history model was estimated on all these documents.

PLSI-PSEUDO: Top n documents from each query unit were concatenated to form a preferred document. The history model was constructed based on topics extracted from these preferred documents.

HISTORY: The history language model was estimated based on all the documents in search history.

TB: It was based on(Tan and Zhai, 2006)which built a unit language model for every past query and the history model was a weighted interpolation of past unit language models.

ORIGINAL: The default search system.

The first 5 systems provided schemes to smooth the query model. They estimated the query models by utilizing different types of feedback (implicit feedback or pseudo feedback) and weighting methods (topic modeling or simple language modeling). The updated query model was an interpolation between MLE query model and history language model. The interpolation parameter was set to 0.5, and n was set to 3.

4.3.3 Performance Comparison

To evaluate the performance on a test query, we focus on two conditions:

1. the test query matches some past interests. We want to check the ability of systems to find relevant information from noisy data.

2. the test query does not match any of past interests. We are interested in the robustness of the systems.

For the first case, the users were asked to select at most 2 queries they submitted for each task. These queries were used as test queries. The other queries were used to simulate the users' search history. In total we got 400 queries for testing. Figure 2 demonstrates the performance of these systems over all test queries. PLSI outperformed all other systems consistently that shows topic model based methods help to estimate a more accurate query model and the user implicit feedback is better evidence. The PLSI-PSEUDO also performed well that indicates the top documents is useful for revealing the topic of queries, even though they do not satisfy user need on occasion. TB also gained better performance than PSEUDO and HISTORY. It indicates

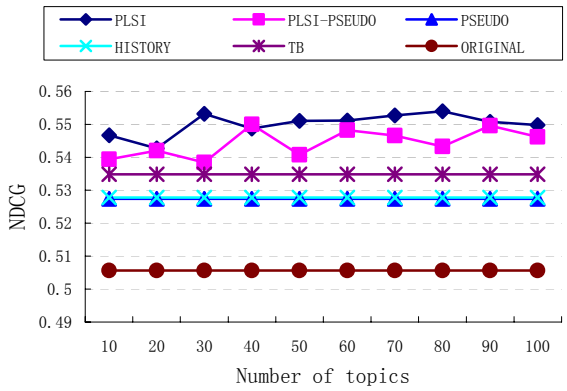

Figure 2: The overall average performance of systems, when each test query matches some user past interests

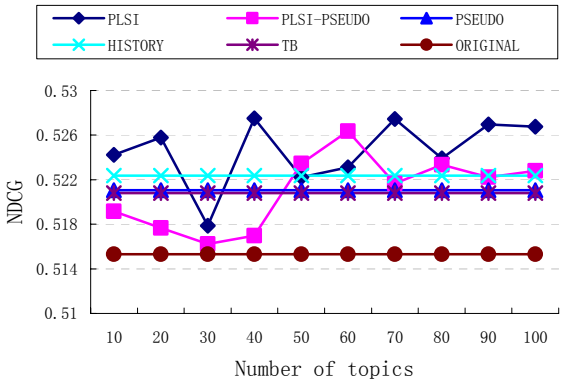

Figure 3: The overall average performance of systems, when each test query does not match any user past interest.

highlighting relevant parts in search history helps to improve the retrieval performance, when the query matches some of user past interests. Compared with default system, both HISTORY and PSEUDO improved a lot which proves that the context in search history is reliable feedback.

For the second case, each user was asked to hold out 5 interests from his collection for testing and the other interests were used as search history. The users selected queries from the held out interests as test queries. These queries did not match each user's past interests. We got 244 test queries. As figure 3 shows, though systems still performed better against ORIGINAL, the improvements were not significant. PLSI still gained the best performance. It has better ability to alleviate the effect of noise. HISTORY and PLSI are more robust than PLSI-PSEUDO which seems sensitive to the number of topics in this case.

In both cases, HISTORY gained moderate performance but quite robust. It is still a very strong baseline, though noisy information is not filtered out. PLSI performed best in both cases. PLSI-PSEUDO outperformed PSEUDO when the test queries matched user past interests and gained comparable results in second case. It shows that modeling user search history as a mixture of topics and weighting topics according to relevance between topics and query help to update a better query model. However, it is necessary to determine if a query matches past interests that helps to optimize personalized search strategies.

5 Conclusion and Future Work

In this paper, we have proposed a topic model based method for personalized search. This approach has some advantages: first, it provides a principled way to combine topic modeling and personalized search; second, it is able to find user preferences in an unsupervised way and gives an informative summary of user search history; third, it explores the underlying relationship between different query units via topics that helps to filter out the noise and improve ranking quality.

In future, we plan to do a large scale study by leveraging the already built search system or business search engines. Also, we will try to add more information to extend the existing model. Besides, it is necessary to design methods for determining whether a submitted query matches the user past interests that is crucial to apply our algorithm adaptively and selectively.

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant No. 60736044, by the National High Technology Research and Development Program of China No. 2008AA01Z144, by Key Laboratory Opening Funding of MOE-Microsoft Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, HIT.KLOF.2009020. We thank the anonymous reviewers and Fikadu Gemechu for their useful comments and help.

References

David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Dempster, A.P., Laird N.M. and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38.

Dou, Z., Su R. and J. Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. *Proc. WWW*, pages 581–590.

Eugene, A., Eric B. and D. Susan. 2005. Improving web search ranking by incorporating user behavior information. *Proc.SIGIR*, pages 19–26.

Eugene, A. and Zijian Zheng. 2006. Identifying best bet web search results by mining past user behavior. *Proc.SIGKDD*, pages 902–908.

Havelivala, T.H. 2002. Topic-sensitive pagerank. *Proc. WWW*, pages 517–526.

Hofmann, T. 1999. Probabilistic latent semantic indexing. *Proc.SIGIR*, pages 50–57.

Janse, B.J., Spink A. Bateman J. and T. Saracevic. 2000. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 26(2):207–222.

Jarvelin, K. and J. Kekakainen. 2000. Ir evaluation methods for retrieving highly relevant documents. *Proc.SIGIR*, pages 41–48.

Lavrenko, V. and W. Croft. 2001. Relevance based language models. *Proc.SIGIR*, pages 120–127.

Pitkow, J., Schutze H. Cass T. Cooley R. Turnbull D. Edmonds A. Adar E. and T. Breuel. 2002. Personalized search. *Commun,ACM*, 45(9):50–55.

Qiu, F. and J. Cho. 2006. Automatic identification of user interest for personalized search. *Proc.WWW*, pages 727–736.

Shen, X., Tan B. and C. Zhai. 2005a. Context-sensitive information retrieval using implicit feedback. *Proc. SIGIR*, pages 43–50.

Shen, X., Tan B. and C. Zhai. 2005b. Implicit user modeling for personalized search. *Proc. CIKM*, pages 824–831.

Silverstein, C., Marais H. Henzinger M. and M. Moricz. 1999. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12.

Speretta, M. and S. Gauch. 2005. Personalized search based on user search histories. *Proc. WI'05*, pages 622–628.

Steyvers, M. and T. Griffiths. 2007. Probabilistic topic models. *Handbook of Latent Semantic Analysis. Erlbaum, Hillsdale, NJ*.

Sugiyama, K., Hatano K. and Yoshkawa. M. 2004. Personalized search based on user search histories. *Proc. WWW*, pages 675–684.

Tan, B., Shen X. and C. Zhai. 2006. Mining long-term search history to improve search accuracy. *Proc.SIGKDD*, pages 718–723.

Tan B., Atulya Velivelli, Fang H. and C. Zhai. 2007. Term feedback for information retrieval with language models. *Proc.SIGIR*, pages 263–270.

Teevan, J., Dumais S.T. and E. Horvitz. 2005. Personalizing search via automated analysis of interests and activities. *Proc.SIGKDD*, pages 449–456.

Teevan, J., Morris M.R. Bush S. 2009. Discovering and using groups to improve personalization. *Proc.WSDM*, pages 15–24.

White, R.W., Bailey P. and L. Chen. 2009. Predicting user interest from contextual information. *Proc.SIGIR*, pages 363–370.

Xu, Jinxi and W. Croft. 1999. Cluster-based language models for distributed retrieval. *Proc.SIGIR*, pages 254–261.

Xu S., Bao, S. Fei B. Su Z. and Y. Yu. 2008. Exploring folksonomy for personalized search. *Proc.SIGIR*, pages 155–162.

Yan, H., Li J. Zhu j. and B. Peng. 2005. Tianwang search engine at trec 2005: Terabyte track. *Proc.TREC*.

Zhai, C. and J. Lafferty. 2001a. A study of smoothing methods for language models applied to ad hoc information retrieval. *Proc.SIGIR*, pages 334–342.

Zhai, Chengxiang and John Lafferty. 2001b. Model-based feedback in the language modeling approach to information retrieval. *Proc.CIKM*, pages 403–410.

Zhai, C., Velivelli A. and B. Yu. 2004. A cross-collection mixture model for comparative text mining. *Proc.SIGKDD*, pages 743–748.

Zhao, Y. and G. Karypis. 2001. Criterion functions for document clustering: Experiments and analysis. *Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN.*