Personalization in Professional Academic Search

Suzan Verberne
Radboud University Nijmegen
S.Verberne@cs.ru.nl

Maya Sappelli
Radboud University Nijmegen / TNO
M.Sappelli@science.ru.nl

Diana Ransgaard Sørensen
The Royal School of Library
and Information Science
D.R.Sorensen@uva.nl

Wessel Kraaij
TNO / Radboud University Nijmegen
W.Kraaij@cs.ru.nl

Abstract

In this paper, we investigated how academic search can profit from personalization by incorporating query history and background knowledge in the ranking of the results. We implemented both techniques in a language modelling framework, using the Indri search engine. For our experiments, we used the iSearch data collection, a large corpus of documents from the physics domain together with 65 search topics from scientists and students. We found that it is possible to improve academic search by taking into account query history. However, we have not been able to prove that terms extracted from the user's background data can improve academic search.

1 Introduction

Professional search is different from ad hoc search in several aspects: it takes place in a specific domain, information is to be found in different types of sources (articles, books and web pages), and the search tasks often have a high complexity. In this paper we focus on *academic* search, which is the professional search carried out by scientists in their daily work. Researchers need to find scientific literature to keep up to date with their field and as a source for writing their own papers.

From the point of view of the search engine, academic information seeking behaviour is often reflected by a sequence of queries, related to one or more specialized topics, intertwined with clicks on scientific papers, books and domain-specific web pages. The user performing the search is a professional in his or her field, which means that he or she has background knowledge on the topic. Given these characteristics of academic information seeking, we argue that the following information retrieval (IR) technologies should at least be integrated in a search engine for academic search: (1) Aggregated search (merging results from different types of sources); (2) IR over query sessions (exploiting query history to improve ranking); (3) Personalization based on background knowledge.

In this paper, we investigate how academic search can profit from incorporating (2) query history and (3) background knowledge in the ranking of the results. We implement both techniques in a language modelling framework, using the Indri search engine.[1] For our experiments, we use the iSearch data collection [LLLI10], a

In: M. Lupu, M. Salampasis, N. Fuhr, A. Hanbury, B. Larsen, H. Strindberg (eds.): Proceedings of the Integrating IR technologies for Professional Search Workshop, Moscow, Russia, 24-March-2013, published at http://ceur-ws.org

[1] http://www.lemurproject.org/indri/

large corpus of documents from the physics domain together with 65 search topics from scientists and students. The collection contains three types of documents (articles, books and metadata), which makes results aggregation necessary. However, the focus of our work is on personalization, not on aggregation.

2 Background and Related work

In this section, we first summarize the work on personalized search and user profiling, which includes work on exploiting query history. Then we will introduce the iSearch collection (Section 2.2), which plays a central role in our work. There is a small body of work using the (relatively new) iSearch collection, which we discuss in Section 2.3.

2.1 Personalized search and user profiling

Many approaches to content-based personalization represent user profiles as a list of **terms from a reference ontology** such as the Open Directory Project (ODP)[2] or Yahoo! directory. For example, in the work by [PG99, GCP03, SG05], the user profile has the form of an ontology in which each term (e.g. the ODP term 'support vector machines') has a weight indicating the user's interest in the concept represented by that term. User data such as previous queries or snippets of visited web pages are automatically mapped to the terminology of the ontology. The resulting profiles are in most studies used to *re-rank* the search results. Alternatively, in the work by [MPS07], the ontology-based approach is exploited to improve a *categorization-based* retrieval system for knowledge workers. In [PG99], an improvement of 8% in terms of 11-point precision is reported as an effect of adding user profiles to result ranking. [SG05] report that the average rank of the results that were selected as relevant by the user improves with 33% by adding profiles that were extracted from query history.

A somewhat similar approach to user profiling is to classify user data in **topical categories**. In [LYM04], a category (e.g. the ODP category 'Artificial Intelligence') is defined as a set of terms with weights, which reflect the significance of the user's interest in that category. In [LYM04], user profiles are learned from the user's search history (queries and click data) and 12%–13% improvement over non-personalized results is obtained. In [QC06], a user's preference is represented as a vector of topics (e.g. 'Computers') with weights. These user preferences are learned from click-history data. The authors implement a personalized search system that extends the PageRank function based on the topics of web pages and the user's preferences for these topics. They find that their personalized method performs 25% to 33% better than Topic-Sensitive PageRank without personalization.

An alternative option for user profile learning is **term extraction**: extracting prominent terms from user data. These terms can then be used for re-ranking search results based on the similarity between the user profile and the retrieved documents (e.g. [MS04]), or for query modification. For example, in [CS98], the query is expanded with the terms from the user profile that are the most correlated to the terms in the query. In [SZ03], a language modelling approach to query modification is followed: a query model is created in which the terms from previous queries are weighted with the terms from the current query. Up to 52% improvement in terms of average precision is achieved when the user's previous three queries are added to the model of the current query. In [STZ05], the current query is expanded with terms from previous queries and from the documents retrieved for those queries. The authors are able to improve over the Google ranking baseline and stress again that their method can improve existing web search performance without any additional effort from the user.

In this paper, we follow a language modelling approach to personalization, incorporating the user's query history and his background knowledge in a merged query model.

2.2 The iSearch collection

For our experiments on academic information seeking behaviour, we use the iSearch data [LLLI10]. The collection consists of 65 natural search tasks (topics) from 23 researchers and students from three different university departments of physics. The search task description form had five fields that the searchers filled in before they started to search for answers:

- a) What are you looking for? (information need)
- b) Why are you looking for this? (work task context)
- c) What is your background knowledge of this topic? (knowledge state)

[2] www.dmoz.org

- d) What should an ideal answer contain to solve your problem or task? (ideal information)
- e) Which central search terms would you use to express your situation and information need? (search terms)

A collection of 18K book records, 144K full text articles and 291K metadata records from the physics field is distributed together with the topics. For each topic, the developers used Indri to collect a pool of up to 200 documents from the collection and the topic owner (also called 'user' or 'searcher' in the remainder of this paper) assessed these documents on their relevance for the topic. Relevance assessments were made on a 4-point scale: highly relevant, fairly relevant, marginally relevant and non-relevant. The average number of topics per searcher is 2.8.

2.3 Previous work with the iSearch collection

All previous work with the iSearch collection uses Indri as index and retrieval engine. As evaluation measure, normalized Discounted Cumulative Gain (nDCG) [JK02] is generally used, because of the graded relevance assessments in the data. An overview of the obtained results in the previous literature is in Table 1.

Table 1: Results previously obtained with the iSearch data set

Paper	Best result (nDCG)	Method
[LKS11] (SIGIR)	0.2161	LM with Jelinek-Mercer smoothing and sense disambiguation; fields a (information need) and e (search terms)
[LLFS12] (SIGIR)	0.2777	LM with Dirichlet smoothing; pseudo-relevance feedback and boosted technical terms
[NdVA12] (TBAS)	0.2890	LM; re-scoring based on inlinks and outlinks
[SBL12] (TBAS)	0.3268	LM with Jelinek-Mercer smoothing (optimized λ, stemming, stopwords filtering); field e (search terms)
[LLI12] (IIiX)	0.3572	LM with Jelinek-Mercer smoothing; fields a (information need) and b (work task context)

[LKS11] compare two types of queries: 'short queries' (fields a and e together) and 'long queries' (all fields a, b, c, d and e together). The results are very similar, short queries giving slightly higher nDCG scores. [LLFS12] calculate the density of technological terms in the documents in the iSearch collection and boost the weight of technical terms in the query. They obtain a significant improvement over the simple retrieval baseline, but a smaller improvement than they achieve with pseudo-relevance feedback. Combining their boosting method with pseudo-relevance feedback gives a small improvement over using pseudo-relevance feedback only. [NdVA12] study the potential of contextualization by re-scoring the Indri result list using inlinks and outlinks of documents. They find that the context from in- and outlinks can help improve retrieval results, but not by a large margin. [SBL12] investigate what gives better results for a collection with different document types (abstracts, papers, books): combining all documents in one index or creating three separate indexes and apply fusion techniques on the result lists. They found no significant difference between the two approaches. The best results so far were obtained by [LLI12] using fields a (information_need) and b (work_task_context) from the topic. They reached an nDCG score of 0.3572.

3 Methodology

3.1 Data preparation

In previous work with the iSearch collection, the terms from the search_terms field (field e) are treated as if they are one single query, sometimes even extended with terms from other fields [LKS11]. Technically, this is not a problem because Indri is "truly best match": not all search terms have to be present in a relevant document. However, we think that concatenating all search terms leads to queries that are unrealistically long for real information seeking behaviour: the average length of the search_terms field is 9.4 words. We noticed that all searchers used punctuation (semicolon, comma or full stop) to structure the search_terms field. The sequences of terms can be interpreted in different ways, and it may be the case that they should be interpreted differently for different users: some searchers seem to have listed multiple facets of the same topic (e.g. "Nano spheres, beads, magnetic, sorting"); others seem to have listed multiple queries that they intended to issue subsequently (e.g. "Electrostatic Force Microscopy (EFM), protein-protein interaction, Avidin-Biotin"). In our inital (baseline) experiments, we consider the search_terms field to be a sequence of queries that were entered within one session. Splitting the search_terms field on the symbols [; , .] leads to an average of 3.5 queries per topic, with an average query length of 2.9 words. This way, we constructed 65 search sessions for 23 searchers with 3.5 queries per session. In Section 4.1 we investigate how we can use the sequence of query terms to improve result ranking.

We indexed the iSearch collection using Indri. We created three separate indexes: one for the 18K book records (BK), one for the 144K full text articles (PF) and one for the 291K metadata records (PN). We did not apply stemming and did not remove stop words.

3.2 Experimental set-up

We used Indri's Language Modelling retrieval function for our retrieval experiments. We found that for our data, Dirichlet smoothing gives better results than Jelinek-Mercer smoothing. Therefore we applied Dirichlet smoothing in our experiments:

$$P(t|D) = \frac{tf_{t,D} + \mu P(t|C)}{|D| + \mu} \tag{1}$$

Here, D represents the document, t is a query term, C represents the collection and $|D|$ is the number of words in the document. $tf_{t,D}$ is the term frequency of t in D. The smoothing factor μ is equal to 2500.

We also compare our results to the pseudo-relevance feedback method that has been implemented in Indri. We tune the parameters for this method using the same grid as [LLFS12]: the number of feedback documents $fbDocs = [1, 2, 5, 10, 20]$ and the number of feedback terms $fbTerms = [3, 5, 10, 20, 40]$. We optimized these parameters for each index separately.

We retrieved a maximum of 1000 results per query, after finding that recall keeps improving when increasing the maximum number of results from 100 up to 1000. We evaluate our results using several different measures, of which the most important are recall and normalized Discounted Cumulative Gain (nDCG) [JK02].

3.3 Aggregating results from different collections

We follow the collection fusion approach described by [SBL12] for merging the results from the three indexes BK (books), PF (articles) and PN (metadata). For each query, we retrieved 1000 results from each of the indexes. We merged the result lists by first normalizing each Indri retrieval score relative to the minimum and maximum scores for that index:

$$score_{norm} = \frac{score_{original} - score_{min}}{score_{max} - score_{min}} \tag{2}$$

We then ranked all documents that are retrieved from the three indexes for a given query by their normalized scores, yielding a combined result list.

4 Implementing personalization in a language modelling framework

The iSearch data give us two different oppurtunities for personalizing search results. The first method is by using the user's query history (Section 4.1) and the second is by using the user's background knowledge as described in their own words (Section 4.2).

4.1 Exploiting query history

We implemented three different methods for exploiting previous queries within a session: (A) Simple concatenation of all queries in the session into one query, thereby simulating a form of query refinement where the initial query is extended with more terms subsequently[3]; (B) Combining results from multiple queries by summing the retrieval scores for each document over all queries for which the document is retrieved, and then rank the documents by their combination scores; (C) Combining query models following the language modelling approach proposed by [SZ03]. In the latter approach, a maximum likelihood estimation is applied to create a unigram language model for each session, assuming a bag of words. This way, a query language model is estimated in

[3]This method makes it possible to compare our results to results in previous work using the iSearch data, in which all search terms were concatenated into one query.

Table 2: The highest ranked 10 terms for three searchers in the iSearch data, extracted from their background knowledge description using the term extraction method described in [TH03]

085	086	087
micro	cavity	biology
flow	medium	background
sorting	equations	microscopy
subject	model	basic
devices	gain	specific
knowledge	element	want
extra knowledge	linear	electrostatic
thesis	limiting element	electrostatic force microscopy
micro particles based	bandwidth	scanning probe
background knowledge	absorber medium	electric

which the terms from all queries from the current session are combined and weighted by the length of the query they occur in:

$$p(t|q_1, ..., q_k) = \frac{1}{k} \sum_{i=1}^{k} \frac{c(t, q_i)}{|q_i|} \tag{3}$$

Here, $c(t, q_i)$ is the number of occurrences of term t in query q_i and $|q_i|$ is the length of query q_i. k is the number of queries in the session. We used the weight operator in the Indri query language to retrieve results for the merged query models: for each session, a query is constructed of the form `#weight(`w_1 t_1 w_2 t_2 ... w_n t_n`)` and submitted to Indri. We evaluate the results per topic (session).

4.2 Personalization based on background knowledge

The average number of topics per searcher in the iSearch data is 2.8. For each topic, the searcher formulated a short text describing his or her background knowledge on the topic. We hypothesize that this background knowledge can be a valuable source for personalization. We exploited the background knowledge descriptions to create profiles of the searchers. In a real information seeking situation, the searcher would not have provided the system with descriptions of his/her background knowledge. Instead, a search system that aims at personalization of retrieval results would use previously accessed documents (online or on the searcher's harddisk) to build a user profile. We therefore consider the background knowledge descriptions as approximations of bigger collections of user data. For modelling the user's knowledge, we concatenated the background_knowledge fields from all topics belonging to one searcher. This results in a set of 24 text documents with an average of 328 words per searcher.[4]

We extracted a user profile from the background knowledge documents in the form of a list of prominent n-grams ($n = 1, 2, 3$). 'Prominence' is determined using the language modelling approach described by [TH03].[5] This method needs a background corpus for determining the prominence of terms in the document. We chose the Corpus of Contemporary American English as background corpus because the developers provide a word frequency list and n-gram frequency lists that are free to download.[6] Note that the same approach could be applied to bigger user collections. The main difference is that the background knowledge descriptions in the iSearch data are relatively sparse because of their limited size. Table 2 shows excerpts from the term lists extracted for the three first searchers in the iSearch data (author ids 085, 086 and 087).

According to [MGSG07], the user profile can affect the search process in three phases: (1) as part of retrieval process (web pages have already been scored according to the user profile before the search takes place); (2) in a re-ranking step (the user profile is used to re-score the documents that have been retrieved using a non-personalized retrieval step); (3) in a query modification step (where the user profile is used to adapt the user's query, after which retrieval and ranking is performed as usual). We chose the third of these possibilities because

[4]We are aware of the fact that the topic-specific background knowledge field might give better results but using this is somewhat less realistic because in reality the searcher does not provide a separate background model for each search topic.

[5]We only used the 'informativeness' criterion from this paper, excluding the 'phraseness' factor from the model because it heavily penalizes unigram terms, while we are interested in unigrams as well as multiword terms.

[6]http://www.wordfrequency.info

it is more efficient than the first option (which requires that a similarity score between the user profile and each document in the collection is calculated before the search starts); we plan to experiment with the second method (re-ranking) in future work.

We incorporated the user profiles in the query results by creating a merged query model, incorporating the top-k ($k = 10$ for the current experiments) terms from the user profile. For the purpose of weighting the terms in the user profile, we used a variant of equation 3, but instead of using the relative counts of each term in all queries ($\sum_{i=1}^{k} \frac{c(t,q_i)}{|q_i|}$), we used the prominence scores from [TH03].

$$p(t|u_k) = \frac{1}{k} prom(t_i, u) \tag{4}$$

Here, k (which was the number of queries in Equation 3) is the number of terms from the user profile that is incorporated in the query. We again used the weight operator in the Indri query language to retrieve results for the merged query model mixed with the user profile model: for each session, a query is constructed of the form "#weight(w_1 t_1 w_2 t_2 ... w_n t_n)", where a term t_i can be a term from the query model or from the user model.

5 Results

The results that we obtained in the retrieval experiments with use of query history are in Table 3. Note that the results for pseudo-relevance feedback were obtained with optimal parameter settings on these data; we did not held out a separate tuning set. The results for personalization with user profiles are in Table 4.

6 Conclusions

In this paper, we investigated how academic search can profit from personalization. We experiment with language modelling methods to incorporate the user's query history and background knowledge in the ranking of the results.

We aggregated the result lists from three different indexes (books, articles and metadata) in one result list. Using the Indri ranking model, we first obtained a baseline nDCG score of 0.231 and using pseudo-relevance feedback an nDCG score of 0.249. Second, we implemented three methods for incorporating query history in the results. With the simplest method (all query terms concatenated) we replicated the results reported in previous research on the same data. The second method, merging the result lists for the subsequent queries after retrieval, yielded the highest recall (63.4%), while the third method (merging query models) resulted in the highest nDCG score (0.364).

Third, we created user profiles by extracting prominent terms from the background knowledge texts formulated by the searchers. We incorporated these profiles in the query model. Unfortunately, we were not able to improve over the results that we obtained without personalization; the highest nDCG that we achieved on the combined indexes was 0.366. We suspect that the sparseness of the user profiles plays an important role. Therefore, in the future, we would like to re-run the experiments with user profiles that are based on more data. In addition, we will experiment with different methods for incorporating the user profile in the result list.

In conclusion, we found that it is possible to improve academic search by taking into account query history. However, we have not been able to prove that terms extracted from the user's background data can improve academic search. More work is needed, and planned, in this direction.

7 Acknowledgements

This publication was supported by the Dutch national program COMMIT (project P7 SWELL).

References

[CS98] L. Chen and K. Sycara. Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–139. ACM, 1998.

[GCP03] S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1(3/4):219–234, 2003.

[JK02] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Table 3: Results for the three separate different indexes and the combined result lists for each subcollections using two baselines and three different methods for exploiting query history (See Section 4.1). For the approaches that exploit query history, the results have been averaged over the 65 topics (sessions); for the setting without any query history, the results have been averaged over all individual queries. Per index, the highest recall and nDCG scores have been marked with boldface.

index	method	session aggregation	# of queries	recall	nDCG
PF	Baseline (Dirichlet smoothing)	none	227	23.8%	0.203
	Pseudo-relevance feedback	none	227	23.9%	0.205
	Baseline (Dirichlet smoothing)	A. queries concatenated	65	45.8%	0.313
	Pseudo-relevance feedback	A. queries concatenated	65	48.1%	**0.315**
	Baseline (Dirichlet smoothing)	B. merged query results	65	**57.6%**	0.170
	Baseline (Dirichlet smoothing)	C. merged query models	65	44.0%	0.313
	Pseudo-relevance feedback	C. merged query models	65	43.9%	0.313
BK	Baseline (Dirichlet smoothing)	none	227	38.6%	0.211
	Pseudo-relevance feedback	none	227	42.5%	0.221
	Baseline (Dirichlet smoothing)	A. queries concatenated	65	74.1%	0.344
	Pseudo-relevance feedback	A. queries concatenated	65	**79.5%**	**0.362**
	Baseline (Dirichlet smoothing)	B. merged query results	65	75.9%	0.224
	Baseline (Dirichlet smoothing)	C. merged query models	65	73.3%	0.332
	Pseudo-relevance feedback	C. merged query models	65	72.2%	0.331
PN	Baseline (Dirichlet smoothing)	none	227	24.3%	0.187
	Pseudo-relevance feedback	none	227	25.2%	0.187
	Baseline (Dirichlet smoothing)	A. queries concatenated	65	57.1%	**0.321**
	Peudo-relevance feedback	A. queries concatenated	65	61.8%	0.320
	Baseline (Dirichlet smoothing)	B. merged query results	65	**66.0%**	0.188
	Baseline (Dirichlet smoothing)	C. merged query models	65	57.0%	0.307
	Pseudo-relevance feedback	C. merged query models	65	59.1%	0.307
Combined	Baseline (Dirichlet smoothing)	none	227	26.0%	0.231
	Pseudo-relevance feedback	none	227	26.9%	0.249
	Baseline (Dirichlet smoothing)	A. queries concatenated	65	54.2%	0.344
	Pseudo-relevance feedback	A. queries concatenated	65	55.7%	0.361
	Baseline (Dirichlet smoothing)	B. merged query results	65	**63.4%**	0.247
	Baseline (Dirichlet smoothing)	C. merged query models	65	53.2%	0.360
	Pseudo-relevance feedback	C. merged query models	65	53.8%	**0.364**

[LKS11] C. Lioma, A. Kothari, and H. Schuetze. Sense discrimination for physics retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1101–1102. ACM, 2011.

[LLFS12] B. Larsen, C. Lioma, I. Frommholz, and H. Schütze. Preliminary study of technical terminology for the retrieval of scientific book metadata records. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1131–1132. ACM, 2012.

[LLI12] Christina Lioma, Birger Larsen, and Peter Ingwersen. Preliminary experiments using subjective logic for the polyrepresentation of information needs. In *Proceedings of the 4th Information Interaction in Context Symposium*, pages 174–183. ACM, 2012.

[LLLI10] Marianne Lykke, Birger Larsen, Haakon Lund, and Peter Ingwersen. Developing a test collection for the evaluation of integrated search. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan Rger, and Keith van Rijsbergen, editors, *Advances in Information Retrieval*, volume 5993 of *Lecture Notes in Computer Science*, pages 627–630. Springer Berlin / Heidelberg, 2010.

Table 4: Results for personalization incorporating user profiles to improve result ranking (see Section 4.2). The retrieval method used is baseline (Dirichlet smoothing). Session results have been combined by merging query models (method C in Table 3).

index	personalization method	# of queries	recall	nDCG
PF	none (merged query model)	65	**44.0%**	0.313
	merged query model with user background model	65	43.8%	**0.317**
BK	none (merged query model)	65	73.3%	0.332
	merged query model with user background model	65	**74.8%**	**0.333**
PN	none (merged query model)	65	57.0%	0.307
	merged query model with user background model	65	**57.4%**	**0.315**
Combined	none (merged query model)	65	53.2%	0.360
	merged query model with user background model	65	**53.4%**	**0.366**

[LYM04] F. Liu, C. Yu, and W. Meng. Personalized web search for improving retrieval effectiveness. *Knowledge and Data Engineering, IEEE Transactions on*, 16(1):28–40, 2004.

[MGSG07] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. Personalized search on the world wide web. *The Adaptive Web*, 4321:195–230, 2007.

[MPS07] Z. Ma, G. Pant, and O.R.L. Sheng. Interest-based personalized search. *ACM Transactions on Information Systems (TOIS)*, 25(1):5, 2007.

[MS04] A. Micarelli and F. Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2):159–200, 2004.

[NdVA12] M.A. Norozi, A.P. de Vries, and P. Arvola. Contextualization from the bibliographic structure. In *Proceedings of the ECIR 2012 Workshop on Task-Based and Aggregated Search (TBAS2012)*, pages 9–13, 2012.

[PG99] A. Pretschner and S. Gauch. Ontology based personalized search. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 391–398. IEEE, 1999.

[QC06] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web*, pages 727–736. ACM, 2006.

[SBL12] D.R. Sørensen, T. Bogers, and B. Larsen. An exploration of retrieval-enhancing methods for integrated search in a digital library. In *Proceedings of the ECIR 2012 Workshop on Task-Based and Aggregated Search (TBAS2012)*, pages 4–8, 2012.

[SG05] M. Speretta and S. Gauch. Personalized search based on user search histories. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 622–628. Ieee, 2005.

[STZ05] X. Shen, B. Tan, and C.X. Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM, 2005.

[SZ03] X. Shen and C.X. Zhai. Exploiting query history for document ranking in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 377–378. ACM, 2003.

[TH03] T. Tomokiyo and M. Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*, pages 33–40. Association for Computational Linguistics, 2003.