Domain Level Personalization Technique

Alessandro Campi
Politecnico di Milano
Dipartimento di Elettronica e Informazione
via Ponzio 34/5, 20133, Milano, Italy
campi@elet.polimi.it

Mirjana Mazuran
Politecnico di Milano
Dipartimento di Elettronica e Informazione
via Ponzio 34/5, 20133, Milano, Italy
mazuran@elet.polimi.it

Stefania Ronchi
Politecnico di Milano
Dipartimento di Elettronica e Informazione
via Ponzio 34/5, 20133, Milano, Italy
ronchi@elet.polimi.it

ABSTRACT

We propose a novel technique for web search personalization that exploits the clustering of the results of web searches. Our approach is based on an automatic characterization of the user search history through the collection of semantic domains and web sources chosen by the user. The semantic domains are the terms extracted directly from the clusters contents which describe the main topics covered by the involved documents. The web sources are the web roots of the urls of the documents. The idea is that a user submits a query to a general purpose search engine and then selects clusters or documents from the resulting list. In the first case we can assume that, for that particular query, the user is interested in the topics covered by the selected clusters. We use these information to construct a user profile by assigning the clusters semantic domains to the query submitted by the user. In the second case we keep trace of the relevance and reliability of web sources of the selected documents, by assigning them to the submitted query.

1. INTRODUCTION

Recent researches investigate the ability of current search engines to address the diverse goals that people have when they submit the same query to a search engine. The potential value of personalizing search results is quantified and great variance was found in the results that different individuals rated as relevant for the same query. The analysis suggests that while search engines perform well in ranking results to maximize global happiness, they do not do a very good job for specific individuals [22]. In recent years a lot of research work was devoted to overcome this limitation by proposing ways to make the search engine aware of the context of its users in order to adapt the search results with respect to it. By learning the context of the users it is possible to personalize the search engine result list and to provide more valuable results to user queries.

In this paper, we present a novel approach of web search personalization, that exploits the clustering of the resulting documents in order to create a complete user profile based on a *characterization* of the user past search history. This operation is realized through the usage of two different information: the *semantic domains*, and the *web sources*. In particular, for *semantic domain* we mean a term s_d that express a "user-specific meaning" of a generic query term q_t (for example if q_t = *Java* we can have s_d = "language" or s_d = "island"). The aim of these domains is to disambiguate a general query term w.r.t. the user preferences. On the other hand, for *web source* we mean the root of the sources considered reliable for the user (i.e. `www.java.com`) w.r.t her/his past searches and document choices. This mechanism of user profile construction has two important characteristics: it is automatically realized without any explicit effort from the user or other contributions from external sources (i.e. ontologies, thesaurus, etc.), and it works at the level of web sites, differently from most of other personalization techniques that act directly on specific documents.

The final goal of our contribution is to learn the preferences of a user in order to support a personalized ranking of future results (both at document and at cluster level), and to provide the user with additional queries which may be interesting with respect to her/his profile.

2. RELATED WORK

Our work is part of a long research stream on personalization in Web searches. The work in [15] proposes a technique to map a user query to a set of categories, which represent the user's search intention. The set of categories are used to disambiguate the query terms while a user profile is learned from the user history and a category hierarchy respectively. This work is different from our in the usage of a predefined set of domains documents are classified in. The Inquirus 2 project [21] uses context information, currently in the form of a category of desired information ("personal bookmarks", "research papers", etc.). Differently from our approach this work does not exploit clustering of pages and the categories used to classify pages are not based on the semantics of their contents.

The work in [3] focuses on re-ranking the Web search output according to the cosine distance between each URL and a set of terms describing user's interests. An evolution of the work is [4] which proposes to improve Web queries by expanding them with terms collected from each user's personal information repository (personal collection of text documents, emails, Web pages, ...). An alternative approach is to compute a topic-oriented PageRank [9], in which PageRank vectors biased on each of the main topics of the Open

VLDB '09, August 24-28, 2009, Lyon, France

Directory were initially calculated off-line, and then combined at run-time based on the similarity between the user query and each of the topics. In this work the set of topics is predefined, while in our approach topics are dynamically built. The idea is extended in [17] by distributing the PageRank across the topics it contains in order to generate topic-oriented rankings. An algorithm that avoids the massive resources needed for storing one personalized PageRank vector per user by precomputing them only for a small set of pages and then applying linear combination is proposed in [11]. Machine Learning on the past click history of the user [19] can be used in order to determine topic preference vectors and then apply Topic-Sensitive PageRank.

Alternative approaches are based on the idea of expanding the user query with new terms related to the input keywords. Such relationships are usually extracted from large scale thesauri, as WordNet [7, 20, 12, 16]. The study in [5] proposes a new method for query expansion based on query logs. The central idea is to extract probabilistic correlations between query terms and document terms by analyzing query logs. These correlations are then used to select high-quality expansion terms for new queries. [13] proposes a method to generate refinements or related terms to queries by mining anchor text for a large hypertext document collection. Search results can be grouped based on different query meanings [10]. It is done using WordNet to determine the meanings of each query term and merging similar senses with a grouping algorithm that employs a combination of categorization and clustering techniques.

Another important research stream is devoted to exploit query expansion in order to obtain a better formulation of user query. The main idea is that useful information can be extracted from the relevant documents returned for the initial query. A literature review of the beginnings of this research topic is described in [6]. [2] introduces the usage of clusters asking users to choose relevant clusters, instead of documents, thus reducing the interaction. Summarization can be used to extract informative sentences from the top-ranked documents and uses these sentences to expand the user query [14]. RIB (Recommender Intelligent Browser) [25] categorizes Web snippets using socially constructed Web directory such as the Open Directory Project. Snippets are clustered to improve the categorization. The created user profile is used to propose search results to users. [8] proposes a method to personalize a user's experience within a folksonomy using clustering. In particular, unsupervised clustering methods are used for extracting commonalities between tags, and the discovered clusters are used as intermediaries between a user's profile and resources in order to tailor the results of search to the user's interests. These approaches are different from ours in the usage of a predefined set of categories: terms are generated to classify pages on-the-fly using the page contents. Similarly, [24] explores an approach that focuses on the "social annotations of the web" which are annotations manually made by normal web users without a pre-defined formal ontology. Compared to the formal annotations, although social annotations are coarse-grained, informal and vague, they are also more accessible to more people and better reflect the meaning of the web resources from the users' point of views during their actual usage of the web resources. The derived emergent semantics are used to discover and search shared web bookmarks.

"London"	
cl.1 Wikipedia (0.669) { <LLondon, 1.0> ; <wikipedia, 0.5> ; <free, 0.5> ; <encyclopedia, 0.33> ; <city, 0.25> }	*London wikipedia*
cl.2 London Trinity College (0.606) { <London, 1> ; <college, 0.5> ; <university, 0.5> ; <king, 0.3> ; <colleges, 0.2> }	*London college*
cl.3 Google News London (0.598) { <London, 1> ; <news, 0.421> ; <city, 0.16> ; <guide, 0.16> ; <times, 0.16> }	*London news*
...	...
cl.8 Travel Guide (0.474) { <London, 1> ; <guide, 0.49> ; <travel, 0.25> ; <theatre, 0.14> ; <entertainment, 0.13 }	*London travel guide*
cl.9 Airport (0.451) { <airport, 1> ; <London, 0.57> ; <international, 0.43> ; <advice, 0.28> ; < parking, 0.28> }	*airport LLondon international*
...	...

Figure 1: Results presentation at clusters level.

3. OUR APPROACH

Our personalization proposal is based on the idea of tracking the choices performed by the user on the list of results obtained after she/he submits a query. The proposed approach exploits Matrioshka [1]. It allows users to submit queries to search engines (such as Google, Yahoo, Google Scholar) in order to obtain clustered and labeled results. Search engines return results representing Web pages characterized by title, url and snippet. When one submits a query to a selected search engine, the resulting documents are clustered using the Lingo clustering algorithm [18]. The result is presented as a list of labeled and ranked clusters. Labels are built considering the set of most relevant terms extracted from titles and snippet of the clustered documents. The retrieved significant terms are also used in order to define a disambiguated query for each cluster. The role of the new queries is to allow deepening the search with more specific queries.

The result of a query submission is shown in Figure 1. In the first column are presented the retrieved clusters. In particular, each cluster c_i is characterized by a label l_i, an overall weight ow_i of relevance of the cluster general content w.r.t. the submitted query, the set of weighted semantic domains $S_i = \{sd_{i1}, ..., sd_{ij}, ..., sd_{in}\}$ associated with the cluster, and their corresponding weights wsd_{ij}. In the second column are presented the disambiguated (expanded) queries extracted from each resulting cluster.

The personalization process is based on the tracking of the choices performed by the user on the result list of clustered documents. Our objective is to learn the user preferences, and to use them in order to:

1. rank the clusters of a search result list, showing in the first positions the clusters containing more interesting contents from the user viewpoint;

2. rank the documents contained in each cluster, in order to show in the first positions inside each cluster the documents belonging to the sources the user preferred in previous search processes;

3. recommend a set of terms taken from her/his profile for the expansion and the specialization of her/his original queries.

To achieve our goals, we consider two types of interaction between the user and the list of the retrieved clusters:

1. *Query execution*: the user chooses to submit a query either by typing it in the search engine or by choosing it in the set of the disambiguated queries associated to clusters or by choosing one of the recommended terms.

2. *Click tracking*: the user chooses a specific document or a specific cluster, in order to explore it.

Both actions indicate that the user is interested in a certain topic. We use these information to create a *user profile*. In particular, we store:

1. the *web sources* ($W = \{ws_1, ws_2, \ldots, ws_n\}$) of the chosen documents (such as `www.expedia.com`, etc). Such information are used to assign a reliability degree to each source. Intuitively, users choose to explore the pages of sources considered reliable, we increase the ranking of the sources users seem to prefer;

2. the *semantic domains* ($S = \{sd_1, sd_2, \ldots, sd_m\}$) considered interesting by the user. Such information are represented by the keywords corresponding to the clusters of interest.

These information are closely related to a specific submitted query and build the history we want to learn.

Differently from the techniques commonly used in personalization, we do not track the information related to a specific document (e.g. url, title, snippet, ...) because such information have a too fine granularity for the personalization process. Furthermore, the usage of the information of single documents often represents a limitation. For example:

1. it is possible to find urls representing different sections of the same web site. To keep trace of all of them, we need to store different information about the same document, obtaining duplicates and wasting space;

2. if we track the exact document url, the stored information could be used in order to represent only that document, and, in particular, only that single page of the entire web site. So, if we retrieve correlated (contiguous) pages as result of the same search, we can bring in the first positions of the resulting list only the pages that has a traced url;

3. the results of different submissions of the same query on a search engine are often different, especially if they were obtained by submissions very distant in time, even if the document sources are usually the same.

Hence, our general assumption is that a user is not actually interested in a specific page of a web site, but she/he is more interested in obtaining documents from a reliable "favorite" source, as first results of a specific query.

So, let us suppose that a computer scientist submits the queries "java" and "apple". We assume that s/he is first interested in obtaining results from her/his more reliable sources such as `www.java.com` and `www.apple.com` respectively, rather then from other sites such as `www.sun.com` or `www.allaboutapple.com/`, or from incoherent sources such as `www.bali-travel-online.com` (that considers java as an island) and `www.applefruit.it` (that considers apple as a fruit). The discrimination on the individual documents is less important under the same query.

4. HISTORY MATRICES

To store all the information needed by the personalization process we propose a data model based on the use of matrices. In particular, we conceive three types of matrices, the User Profile matrix, the Source Reputation matrix and the Source Annotation matrix. In this section we give the details about each of them.

4.1 User Profile matrix

The User Profile matrix P is constructed from the user query terms and the semantic domains associated with the clusters the user has clicked on. Given a query Q composed of a set of terms $\{qt_1, qt_2, ..., qt_n\}$, every time the user clicks on a cluster c, identified by the semantic domains in S, it means that the user is interested in the semantic domains in S, with respect to the query Q. This consideration is used to update the User Profile matrix which associates a degree of relevance between query terms and semantic domains. In fact, the matrix represents the function $\mathcal{P} : \{qt_i, sd_j\} \rightarrow w_{ij}$, which associates with each query term qt_i and each semantic domain sd_j a weight w_{ij} which indicates how relevant sd_j is with respect to qt_i.

In particular, the weight w_{ij} of a couple $\{qt_i, sd_j\}$ is the average of the weights of sd_j in their original clusters. Such average is computed with respect to the number of times the user has clicked on a cluster containing sd_j among its semantic domains. Moreover, in the matrix each query term is coupled with the number of times the user has asked for a query containing that term, and each semantic domain is coupled with the number of times the user has been interested in that semantic domain (which means s/he has clicked on a cluster containing those keywords). The matrix is represented as:

	$\{sd_1,f_1\}$	...	$\{sd_m,f_m\}$
$\{qt_1,f_1\}$			
...		w_{ij}	
$\{qt_n,f_n\}$			

where:

$$w_{ij_{new}} = \frac{w_{ij_{old}} * f_{j_{old}} + w_{j_c}}{f_{j_{new}}},$$

is computed incrementally using the weights of the semantic domain (sd_j) obtained for the same query term (qt_i) but in various submission, and $f_{j_{new}} = f_{j_{old}} + 1$ is the new frequency value associated with sd_j.

Let us suppose the user queries "London" for the first time and obtains as a result the following two clusters (for each cluster is indicated its set of semantic domains and the weight each of them has in the cluster):

```
c1: {(hotel,0.4),(travel,0.6)}
c2: {(theater,0.2),(movie,0.3),(entertainment,0.5)}
```

the user then clicks on the second cluster in order to examine its content. This action would result in the following User Profile matrix:

	{theater,1}	{movie,1}	{entertainment,1}
{london,1}	0.2	0.3	0.5

Let us now suppose the user queries "London hotels" and obtains the following clusters:

```
c3: {(flight,0.4),(travel,0.6),(entertainment,0.3)}
c4: {(economic,0.2),(booking,0.3)}
```

and then clicks on the first cluster. The new User Profile matrix is[1]:

	{th,1}	{mo,1}	{en,2}	{fl,1}	{tr,1}
{london,2}	0.2	0.3	0.4	0.4	0.6
{hotel,1}	0	0	0.3	0.4	0.6

The weight of the semantic domain "entertainment" for the query term "london" is evaluated as $(0.5 + 0.3)/2 = 0.4$.

Note that if we compute the logic AND between the query term "london" row and query term "hotel" row, we can easily identify which semantic domains are related to the multi-word query "london hotel" (in this case: flights, travel and entertainment). Thus, this matrix is suitable for both single-term and multi-term queries.

4.2 Source reputation matrix

The Source Reputation matrix R is constructed from the query terms and the web sources of the documents the user has clicked on. Given a query Q, we assume that every time the user clicks on a document, s/he is interested in that specific source of information with respect to the query. This consideration is used to update the source reputation matrix which associates a frequency between query terms and web sources. In fact, the matrix represents the function $\mathcal{R} : \{qt_i, ws_k\} \rightarrow f_{ik}$, which associates with each query term qt_i and each web source ws_k a frequency f_{ik} which is the number of times the user has clicked on a document whose source is ws_k, with respect to the query. The matrix is represented as:

	ws_1	...	ws_k
qt_1			
...		f_{ik}	
qt_n			

Let us suppose the user queries "London" and then clicks on the following documents:

```
d1: en.wikipedia.org/wiki/London
d2: www.visitlondon.com/
d3: www.expedia.co.uk/
```

Subsequently, the user queries "London hotels", and then clicks on the following documents:

```
d4: http://www.visitlondon.com/accommodation/
          hotels/
d5: http://www.expedia.co.uk/daily/holidays/
          packages.aspx?rfrr=-13006
```

The final Source Reputation Matrix is[2]:

	wiki	visitlondon	expedia
london	1	2	2
hotel	0	1	1

[1]In the following table we use some shortcuts: th=theater, mo=movie, en=entertainment, fl=flight, tr=travel

[2]In the following table we use some shortcuts: wiki=en.wikipedia.org, visitlondon=www.visitlondon.com, and expedia=www.expedia.co.uk

As for the previous matrix P, also the structure of R allows the identification of relevant multi-terms query sources through the application of the AND operator on the involved query terms rows. Thus `en.wikipedia.org` is not a relevant source for the query "London hotels" because the AND between 1 (for the corresponding "london" row) and 0 (for the corresponding "hotel" row) returns 0. Together with the matrix we retain a value called MAX_R devoted to store the maximum number contained in the matrix R. This is useful for normalization purposes.

4.3 Source Annotation matrix

The Source Annotation matrix A is constructed as the multiplication between the data in the User Profile matrix P and the data in the Source Reputation matrix S, such as $A = P^T \times S$. In this way, the source annotation matrix is used to create a relation between the web sources and the semantic domains. In fact, the matrix represents the function $\mathcal{A} : \{sd_j, ws_k\} \rightarrow w_{jk}$, which assigns to each semantic domain sd_j and each web source ws_k a weight w_{jk} that indicates how much a semantic domain is relevant with respect to a web sources. The matrix is represented as:

	ws_1	...	ws_k
sd_1			
...		w_{jk}	
sd_m			

Let us consider the two examples presented in the previous sections. The resulting Source Annotation Matrix is:

	wiki	visitlondon	expedia
theater	0,2	0,4	0,4
movie	0,3	0,6	0,6
entertainment	0,4	1,1	1,1
flight	0,4	1,2	1,2
travel	0,6	1,8	1,8

5. USER PROFILE CONSTRUCTION AND MAINTENANCE

As already introduced in Section 3, the information we consider in our personalization technique can be collected and updated when:

- the user submits a query;
- the user chooses a certain cluster. We can deduce that the set of semantic domains corresponding to that cluster are of interest to the user, with respect to the query;
- the user chooses a certain document. We can deduce that the web source of the document is considered reliable by the user w.r.t the query.

When the user submits a query, we use the historical data to rank the results of the query and to suggest the user new possibly interesting queries. On the other hand, when the user performs some choices on the results of a query, we use such information to update the historical data.

In the following we give a more detailed description of all the operations we considered interesting for the user profile construction and maintenance.

5.1 Query execution

The execution of a query happens in three different scenarios: (1) the user submits her/his query to the search engine; (2) the user submits one of the possible disambiguated queries built considering her/his profile; (3) the user submits one disambiguated query from the search results list and has therefore submitted the new request.

The query is then first preprocessed by the engine which means that all stop-words are removed and all terms are stemmed. The result is the set of significant terms contained in the query, $Q = \{qt_1, qt_2, \ldots, qt_n\}$. Each term is then used to update the history matrices P and R. In particular, $\forall$ terms $qt_i \in Q$:

- if qt_i is already contained within the P matrix, its frequency is increased by one and the corresponding value of f is updated
- otherwise, it is added to it with $f = 1$
- if qt_i is not contained within the R matrix, it is necessary to add a new line representing the new term

5.2 Choice of a cluster

The user chooses a particular cluster c_j and explores the documents contained in the cluster. In this scenario, the set of semantics domains associated with the cluster is used to update the User Profile matrix P. In particular, $\forall$ term $qt_i \in Q$, the semantic domains $S = \{sd_1, sd_2, \ldots, sd_m\}$ associated with the cluster c_j, are eventually added as columns of P (if they don't already exist in it). Moreover, $\forall w_{ij} \in P$ such as $i = 0, ..., n$ are indexes of query terms which originated the selected cluster:

- if $w_{ij} \neq 0$ in P, its value is updated with the sd_j weight in the selected cluster, as described in 4.1.
- if $w_{ij} = 0$ in P, $w_{ij} = sd_j$ weight.

5.3 Choice of a document

The user chooses a specific document d_h contained in a cluster c_l. In this scenario, the web source wd_j of the document is used to update the Source Reputation matrix R.

In particular, $\forall$ term $qt_i \in Q$:

- a corresponding row in the R table is created (if it is not already present in the matrix)
- a column for web source ws_j is added (if it is not already present in the matrix)
- the frequency f_{ij} is evaluated for the web source ws_j. In particular:
  - if the web source is already contained in the set, its frequency f_{ij} is added by 1.
  - if the web source is not contained, it is added to it with $f_{ij} = 1$.

Note that in this scenario the semantic domains associated to the explored cluster are not taken into account. In fact they were already stored when the user clicked on the cluster label, looking for interesting documents.

In order to avoid the explosion of the dimensions of the matrices and guarantee the scalability of the approach we use classical techniques for efficient sparse matrices management and cleaning techniques to delete not useful data.

6. RANKING OF THE RESULTS

Let us suppose the user submits a query. The result of the query is a set of clustered documents which need to be ranked before being presented to the user. In order to rank the results, we access the historical matrices with the aim of giving a higher rank to the information "similar" to the previously browsed. Algorithm 1 shows how to rank the clusters resulting from a web search given the user profile matrix P, the set of clusters C and the query Q submitted by the user.

Algorithm 1 Rank-Clusters (P, C, Q)

1: **for all** clusters $c_j \in C$ **do**
2: K=\{semantic domain $sd_k \in c_j$\}
3: for all qt consider the set ROW of rows of matrix P corresponding to the query terms
4: build the set COL of columns having only non zero values in the cells in the rows in R
5: insert in S_Q the terms corresponding to COL
6: $X = \{wsd_{ij} : sd_{ij} \in (K \cap S_Q)\}$
7: $CW = \frac{\sum_X x}{|K \cap S_Q|}$
8: $Y = \{wsd_{ij} : sd_{ij} \in S_Q\}$
9: $PW = \frac{\sum_Y y}{|S_Q|}$
10: $P = \frac{CW + PW}{2}$
11: new_rank(c_j)=(1-β)*original_rank(c_j)+β*P
12: Rank-Documents(c_j, H, Q)
13: **end for**
14: OrderByRank(C)
15: **return** C

where β is the personalization factor. As in the work proposed in [23] β is used to decide the weight of the personalization in the computation of the rank varying from 0 to 1. We allow the user to choose the value of β in order to decide how much the desired results are to be near to her/his profile. If β is 0, the ranking is the same as plain search. If β is 1.0, then the search rank is totally determined by the profile. If β is 0.5, which is the default, the system considers equally the importance of the two contributions. In this algorithm CW (content weight) represents the weight of the semantic domains in the cluster which are also contained in the user profile w.r.t. the query. PW (profile weight) represents the weight given to the semantic domains w.r.t. to the terms of the query in the user profile.

7. PROFILE BASED QUERY DISAMBIGUATION

In addition to the query expanded using terms derived from clusters, we want to offer a set of user profile based disambiguated queries. In particular we show a set of terms taken from the semantic domains stored inside the user personal profile, highlighting them w.r.t. their frequency values stored in the User Profile matrix. The user can select one or more of these terms in order to build a new query that is submitted to the search engine. The main difference between the query built considering terms taken from clusters and the queries built using terms taken from the profile is that the first queries propose new original contents that are not correlated with the user's preferences while the others are closer to what the user usually searches while the others.

Algorithm 2 Rank-Documents (R, c, Q)

1: **for all** documents $d_k \in c$ **do**
2: for all qt consider the set ROW of rows of matrix R corresponding to the query terms
3: build the set COL of columns having only non zero values in the cells in the rows in R
4: insert in WS_Q the terms corresponding to COL
5: ws_k=web_source(d_k)
6: **if** $ws_k \in WS_Q$ **then**
7: assign to f the value in R corresponding to ws_k and qt
8: $WP = \frac{f}{MAX\_R}$
9: new_rank(d_k) = (1-β)*original_rank(d_k) + β*WP
10: **end if**
11: **end for**
12: OrderByRank(c)

8. CONCLUSIONS

In this paper we proposed a novel personalization technique based on the extraction of user-preferences information from the clustering of the user web searches results. The basic idea of the proposed approach is to store for each user information about the preferred semantic domains Web sources considered more reliable. The collected information are used to build a user profile useful to re-rank the results in order to offer the higher positions the results with a higher degree of semantic correlation with the user profile and originating from the more reliable Web sources.

9. REFERENCES

[1] G. Bordogna, A. Campi, G. Psaila, and S. Ronchi. A language for manipulating clustered web documents results. In *CIKM '08*, pages 23–32, New York, NY, USA, 2008. ACM.

[2] C.-H. Chang and C.-C. Hsu. Integrating query expansion and conceptual relevance feedback for personalized web information retrieval. *Comput. Netw. ISDN Syst.*, 30(1-7):621–623, 1998.

[3] P.-A. Chirita, C. S. Firan, and W. Nejdl. Summarizing local context to personalize global web search. In *CIKM '06*, pages 287–296, New York, NY, USA, 2006. ACM.

[4] P. A. Chirita, C. S. Firan, and W. Nejdl. Personalized query expansion for the web. In *SIGIR '07*, pages 7–14, New York, NY, USA, 2007. ACM.

[5] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *WWW '02*, pages 325–332, New York, NY, USA, 2002. ACM.

[6] E. N. Efthimiadis. User choices: a new yardstick for the evaluation of ranking algorithms for interactive query expansion. *Inf. Process. Manage.*, 31(4):605–620, 1995.

[7] Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[8] J. Gemmell, A. Shepitsen, M. Mobasher, and R. Burke. Personalization in folksonomies based on tag clustering. In *Proc. of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, July 2008.

[9] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW '02*, pages 517–526, New York, NY, USA, 2002. ACM.

[10] R. Hemayati, W. Meng, and C. T. Yu. Semantic-based grouping of search engine results using wordnet. In *APWeb/WAIM*, pages 678–686, 2007.

[11] G. Jeh and J. Widom. Scaling personalized web search. In *WWW '03*, pages 271–279, New York, NY, USA, 2003. ACM.

[12] S.-B. Kim, H.-C. Seo, and H.-C. Rim. Information retrieval using word senses: root sense tagging approach. In *SIGIR '04*, pages 258–265, New York, NY, USA, 2004. ACM.

[13] R. Kraft and J. Zien. Mining anchor text for query refinement. In *WWW '04*, pages 666–674, New York, NY, USA, 2004. ACM.

[14] A. M. Lam-Adesina and G. J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In *SIGIR '01*, pages 1–9, New York, NY, USA, 2001. ACM.

[15] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *CIKM '02*, 2002.

[16] S. Liu, F. Liu, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *SIGIR '04*, pages 266–272, New York, NY, USA, 2004. ACM.

[17] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *SIGIR '06*, pages 91–98, New York, NY, USA, 2006. ACM.

[18] S. Osinski, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Systems*, pages 359–368, 2004.

[19] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *WWW '06*, pages 727–736, New York, NY, USA, 2006. ACM.

[20] C. Shah and W. B. Croft. Evaluating high accuracy retrieval techniques. In *SIGIR '04*, pages 2–9, New York, NY, USA, 2004. ACM.

[21] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04*, 2004.

[22] J. Teevan, S. T. Dumais, and E. Horvitz. Characterizing the value of personalizing search. In *SIGIR '07*, pages 757–758, New York, NY, USA, 2007. ACM.

[23] J. wook Ahn, P. Brusilovsky, D. He, J. Grady, and Q. Li. Personalized web exploration with task models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1–10, New York, NY, USA, 2008. ACM.

[24] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *WWW '06*, pages 417–426, New York, NY, USA, 2006. ACM.

[25] D. Zhu and H. Dreher. Improving web search by categorization, clustering, and personalization. In *ADMA '08*, pages 659–666, Berlin, Heidelberg, 2008. Springer-Verlag.