Personalized Query Expansion for the Web

Paul - Alexandru Chirita
L3S Research Center*
Appelstr. 9a
30167 Hannover, Germany
chirita@l3s.de

Claudiu S. Firan
L3S Research Center
Appelstr. 9a
30167 Hannover, Germany
firan@l3s.de

Wolfgang Nejdl
L3S Research Center
Appelstr. 9a
30167 Hannover, Germany
nejdl@l3s.de

ABSTRACT

The inherent ambiguity of short keyword queries demands for enhanced methods for Web retrieval. In this paper we propose to improve such Web queries by expanding them with terms collected from each user's Personal Information Repository, thus implicitly personalizing the search output. We introduce five broad techniques for generating the additional query keywords by analyzing user data at increasing granularity levels, ranging from term and compound level analysis up to global co-occurrence statistics, as well as to using external thesauri. Our extensive empirical analysis under four different scenarios shows some of these approaches to perform very well, especially on ambiguous queries, producing a very strong increase in the quality of the output rankings. Subsequently, we move this personalized search framework one step further and propose to make the expansion process adaptive to various features of each query. A separate set of experiments indicates the adaptive algorithms to bring an additional statistically significant improvement over the best static expansion approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms, Experimentation, Measurement

Keywords

Personalized Web Search, Query Expansion, Desktop Profile, Keyword Extraction, Keyword Co-occurrences

1. INTRODUCTION

The booming popularity of search engines has determined simple keyword search to become the only widely accepted user interface for seeking information over the Web. Yet keyword queries are inherently ambiguous. The query "canon book" for example covers several different areas of interest: religion, photography, literature, and music. Clearly, one would prefer search output to be aligned with user's topic(s) of interest, rather than displaying a selection of popular URLs from each category. Studies have shown that more than 80% of the users would prefer to receive such personalized search results [33] instead of the currently generic ones.

Query expansion assists the user in formulating a better query, by appending additional keywords to the initial search request in order to encapsulate her interests therein, as well as to focus the Web search output accordingly. It has been shown to perform very well over large data sets, especially with short input queries (see for example [19, 3]). This is exactly the Web search scenario!

In this paper we propose to enhance Web query reformulation by exploiting the user's Personal Information Repository (PIR), i.e., the personal collection of text documents, emails, cached Web pages, etc. Several advantages arise when moving Web search personalization down to the Desktop level (note that by "Desktop" we refer to PIR, and we use the two terms interchangeably). First is of course the quality of personalization: The local Desktop is a rich repository of information, accurately describing most, if not all interests of the user. Second, as all "profile" information is stored and exploited locally, on the personal machine, another very important benefit is privacy. Search engines should not be able to know about a person's interests, i.e., they should not be able to connect a specific person with the queries she issued, or worse, with the output URLs she clicked within the search interface[1] (see Volokh [35] for a discussion on privacy issues related to personalized Web search).

Our algorithms expand Web queries with keywords extracted from user's PIR, thus implicitly personalizing the search output. After a discussion of previous works in Section 2, we first investigate the analysis of local Desktop query context in Section 3.1.1. We propose several keyword, expression, and summary based techniques for determining expansion terms from those personal documents matching the Web query best. In Section 3.1.2 we move our analysis to the global Desktop collection and investigate expansions based on co-occurrence metrics and external thesauri. The experiments presented in Section 3.2 show many of these approaches to perform very well, especially on ambiguous queries, producing NDCG [15] improvements of up to 51.28%. In Section 4 we move this algorithmic framework further and propose to make the expansion process adaptive to the clarity level of the query. This yields an additional improvement of 8.47% over the previously identified best algorithm. We conclude and discuss further work in Section 5.

*Part of this work was performed while the author was visiting Yahoo! Research, Barcelona, Spain.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

[1]Search engines can map queries at least to IP addresses, for example by using cookies and mining the query logs. However, by moving the user profile at the Desktop level we ensure such information is not explicitly associated to a particular user and stored on the search engine side.

2. PREVIOUS WORK

This paper brings together two IR areas: Search Personalization and Automatic Query Expansion. There exists a vast amount of algorithms for both domains. However, not much has been done specifically aimed at combining them. In this section we thus present a separate analysis, first introducing some approaches to personalize search, as this represents the main goal of our research, and then discussing several query expansion techniques and their relationship to our algorithms.

2.1 Personalized Search

Personalized search comprises two major components: (1) User profiles, and (2) The actual search algorithm. This section splits the relevant background according to the focus of each article into either one of these elements.

Approaches focused on the User Profile. Sugiyama et al. [32] analyzed surfing behavior and generated user profiles as features (terms) of the visited pages. Upon issuing a new query, the search results were ranked based on the similarity between each URL and the user profile. Qiu and Cho [26] used Machine Learning on the past click history of the user in order to determine topic preference vectors and then apply Topic-Sensitive PageRank [13]. User profiling based on browsing history has the advantage of being rather easy to obtain and process. This is probably why it is also employed by several industrial search engines (e.g., Yahoo! MyWeb[2]). However, it is definitely not sufficient for gathering a thorough insight into user's interests. More, it requires to store all personal information at the server side, which raises significant privacy concerns.

Only two other approaches enhanced Web search using Desktop data, yet both used different core ideas: (1) Teevan et al. [34] modified the query *term weights* from the BM25 weighting scheme to incorporate user interests as captured by their Desktop indexes; (2) In Chirita et al. [6], we focused on *re-ranking* the Web search output according to the cosine distance between each URL and a set of Desktop terms describing user's interests. Moreover, none of these investigated the adaptive application of personalization.

Approaches focused on the Personalization Algorithm. Effectively building the personalization aspect directly into PageRank [25] (i.e., by biasing it on a target set of pages) has received much attention recently. Haveliwala [13] computed a topic-oriented PageRank, in which 16 PageRank vectors biased on each of the main topics of the Open Directory were initially calculated off-line, and then combined at run-time based on the similarity between the user query and each of the 16 topics. More recently, Nie et al. [24] modified the idea by distributing the PageRank of a page across the topics it contains in order to generate topic oriented rankings. Jeh and Widom [16] proposed an algorithm that avoids the massive resources needed for storing one Personalized PageRank Vector (PPV) per user by precomputing PPVs only for a small set of pages and then applying linear combination. As the computation of PPVs for larger sets of pages was still quite expensive, several solutions have been investigated, the most important ones being those of Fogaras and Racz [12], and Sarlos et al. [30], the latter using rounding and count-min sketching in order to fastly obtain accurate enough approximations of the personalized scores.

2.2 Automatic Query Expansion

Automatic query expansion aims at deriving a better formulation of the user query in order to enhance retrieval. It is based on exploiting various social or collection specific characteristics in order to generate additional terms, which are appended to the original input keywords before identifying the matching documents returned as output. In this section we survey some of the representative query expansion works grouped according to the source employed to generate additional terms: (1) Relevance feedback, (2) Collection based co-occurrence statistics, and (3) Thesaurus information. Some other approaches are also addressed in the end of the section.

Relevance Feedback Techniques. The main idea of Relevance Feedback (RF) is that useful information can be extracted from the relevant documents returned for the initial query. First approaches were manual [28] in the sense that the user was the one choosing the relevant results, and then various methods were applied to extract new terms, related to the query and the selected documents. Efthimiadis [11] presented a comprehensive literature review and proposed several simple methods to extract such new keywords based on term frequency, document frequency, etc. We used some of these as inspiration for our Desktop specific techniques. Chang and Hsu [5] asked users to choose relevant clusters, instead of documents, thus reducing the amount of interaction necessary. RF has also been shown to be effectively automatized by considering the top ranked documents as relevant [37] (this is known as Pseudo RF). Lam and Jones [21] used summarization to extract informative sentences from the top-ranked documents, and appended them to the user query. Carpineto et al. [4] maximized the divergence between the language model defined by the top retrieved documents and that defined by the entire collection. Finally, Yu et al. [38] selected the expansion terms from vision-based segments of Web pages in order to cope with the multiple topics residing therein.

Co-occurrence Based Techniques. Terms highly co-occurring with the issued keywords have been shown to increase precision when appended to the query [17]. Many statistical measures have been developed to best assess "term relationship" levels, either analyzing entire documents [27], lexical affinity relationships [3] (i.e., pairs of closely related words which contain exactly one of the initial query terms), etc. We have also investigated three such approaches in order to identify query relevant keywords from the rich, yet rather complex Personal Information Repository.

Thesaurus Based Techniques. A broadly explored method is to expand the user query with new terms, whose meaning is closely related to the input keywords. Such relationships are usually extracted from large scale thesauri, as WordNet [23], in which various sets of synonyms, hypernyms, etc. are predefined. Just as for the co-occurrence methods, initial experiments with this approach were controversial, either reporting improvements, or even reductions in output quality [36]. Recently, as the experimental collections grew larger, and as the employed algorithms became more complex, better results have been obtained [31, 18, 22]. We also use WordNet based expansion terms. However, we base this process on analyzing the Desktop level relationship between the original query and the proposed new keywords.

Other Techniques. There are many other attempts to extract expansion terms. Though orthogonal to our approach, two works are very relevant for the Web environment: Cui et al. [8] generated word correlations utilizing the probability for query terms to appear in each document, as computed over the search engine logs. Kraft and Zien [19] showed that anchor text is very similar to user queries, and thus exploited it to acquire additional keywords.

3. QUERY EXPANSION USING DESKTOP DATA

Desktop data represents a very rich repository of profiling information. However, this information comes in a very unstructured way, covering documents which are highly diverse in for-

[2]http://myWeb2.search.yahoo.com

mat, content, and even language characteristics. In this section we first tackle this problem by proposing several lexical analysis algorithms which exploit user's PIR to extract keyword expansion terms at various granularities, ranging from term frequency within Desktop documents up to utilizing global co-occurrence statistics over the personal information repository. Then, in the second part of the section we empirically analyze the performance of each approach.

3.1 Algorithms

This section presents the five generic approaches for analyzing user's Desktop data in order to provide expansion terms for Web search. In the proposed algorithms we gradually increase the amount of personal information utilized. Thus, in the first part we investigate three local analysis techniques focused only on those Desktop documents matching user's query best. We append to the Web query the most relevant terms, compounds, and sentence summaries from these documents. In the second part of the section we move towards a global Desktop analysis, proposing to investigate term co-occurrences, as well as thesauri, in the expansion process.

3.1.1 Expanding with Local Desktop Analysis

Local Desktop Analysis is related to enhancing Pseudo Relevance Feedback to generate query expansion keywords from the PIR best hits for user's Web query, rather than from the top ranked Web search results. We distinguish three granularity levels for this process and we investigate each of them separately.

Term and Document Frequency. As the simplest possible measures, TF and DF have the advantage of being very fast to compute. Previous experiments with small data sets have showed them to yield very good results [11]. We thus independently associate a score with each term, based on each of the two statistics. The TF based one is obtained by multiplying the actual frequency of a term with a position score descending as the term first appears closer to the end of the document. This is necessary especially for longer documents, because more informative terms tend to appear towards their beginning [10]. The complete TF based keyword extraction formula is as follows:

$$TermScore = \left[\frac{1}{2} + \frac{1}{2} \cdot \frac{nrWords - pos}{nrWords}\right] \cdot log(1 + TF) \quad (1)$$

where $nrWords$ is the total number of terms in the document and pos is the position of the first appearance of the term; TF represents the frequency of each term in the Desktop document matching user's Web query.

The identification of suitable expansion terms is even simpler when using DF: Given the set of Top-K relevant Desktop documents, generate their snippets as focused on the original search request. This query orientation is necessary, since the DF scores are computed at the level of the entire PIR and would produce too noisy suggestions otherwise. Once the set of candidate terms has been identified, the selection proceeds by ordering them according to the DF scores they are associated with. Ties are resolved using the corresponding TF scores.

Note that a hybrid TFxIDF approach is not necessarily efficient, since one Desktop term might have a high DF on the Desktop, while being quite rare in the Web. For example, the term "PageRank" would be quite frequent on the Desktop of an IR scientist, thus achieving a low score with TFxIDF. However, as it is rather rare in the Web, it would make a good resolution of the query towards the correct topic.

Lexical Compounds. Anick and Tipirneni [2] defined the *lexical dispersion hypothesis*, according to which an expression's lexical dispersion (i.e., the number of different compounds it appears in within a document or group of documents) can be used to automatically identify key concepts over the input document set. Although several possible compound expressions are available, it has been shown that simple approaches based on noun analysis are almost as good as highly complex part-of-speech pattern identification algorithms [1]. We thus inspect the matching Desktop documents for all their lexical compounds of the following form:

$$\{\ adjective?\ noun+\ \}$$

All such compounds could be easily generated off-line, at indexing time, for all the documents in the local repository. Moreover, once identified, they can be further sorted depending on their dispersion within each document in order to facilitate fast retrieval of the most frequent compounds at run-time.

Sentence Selection. This technique builds upon sentence oriented document summarization: First, the set of relevant Desktop documents is identified; then, a summary containing their most important sentences is generated as output. Sentence selection is the most comprehensive local analysis approach, as it produces the most detailed expansions (i.e., sentences). Its downside is that, unlike with the first two algorithms, its output cannot be stored efficiently, and consequently it cannot be computed off-line. We generate sentence based summaries by ranking the document sentences according to their salience score, as follows [21]:

$$SentenceScore = \frac{SW^2}{TW} + PS + \frac{TQ^2}{NQ}$$

The first term is the ratio between the square amount of significant words within the sentence and the total number of words therein. A word is significant in a document if its frequency is above a threshold as follows:

$$TF > ms = \begin{cases} 7 - 0.1 * (25 - NS) & , if\ NS < 25 \\ 7 & , if\ NS \in [25, 40] \\ 7 + 0.1 * (NS - 40) & , if\ NS > 40 \end{cases}$$

with NS being the total number of sentences in the document (see [21] for details). The second term is a position score set to $(Avg(NS) - SentenceIndex)/Avg^2(NS)$ for the first ten sentences, and to 0 otherwise, $Avg(NS)$ being the average number of sentences over all Desktop items. This way, short documents such as emails are not affected, which is correct, since they usually do not contain a summary in the very beginning. However, as longer documents usually do include overall descriptive sentences in the beginning [10], these sentences are more likely to be relevant. The final term biases the summary towards the query. It is the ratio between the square number of query terms present in the sentence and the total number of terms from the query. It is based on the belief that the more query terms contained in a sentence, the more likely will that sentence convey information highly related to the query.

3.1.2 Expanding with Global Desktop Analysis

In contrast to the previously presented approach, global analysis relies on information from across the entire personal Desktop to infer the new relevant query terms. In this section we propose two such techniques, namely term co-occurrence statistics, and filtering the output of an external thesaurus.

Term Co-occurrence Statistics. For each term, we can easily compute off-line those terms co-occurring with it most frequently in a given collection (i.e., PIR in our case), and then exploit this information at run-time in order to infer keywords highly correlated with the user query. Our generic co-occurrence based query expansion algorithm is as follows:

Algorithm 3.1.2.1. Co-occurrence based keyword similarity search.

Off-line computation:
1: **Filter** potential keywords k with $DF \in [10, \ldots, 20\% \cdot N]$
2: **For** each keyword k_i
3: **For** each keyword k_j
4: Compute SC_{k_i,k_j}, the similarity coefficient of (k_i, k_j)

On-line computation:
1: **Let** S be the set of keywords,
potentially similar to an input expression E.
2: **For** each keyword k of E:
3: $S \leftarrow S \cup TSC(k)$, where $TSC(k)$ contains the
Top-K terms most similar to k
4: **For** each term t of S:
5a: **Let** $Score(t) \leftarrow \prod_{k \in E}(0.01 + SC_{t,k})$
5b: **Let** $Score(t) \leftarrow \#DesktopHits(E|t)$
6: **Select** Top-K terms of S with the highest scores.

The off-line computation needs an initial trimming phase (step 1) for optimization purposes. In addition, we also restricted the algorithm to computing co-occurrence levels across nouns only, as they contain by far the largest amount of conceptual information, and as this approach reduces the size of the co-occurrence matrix considerably. During the run-time phase, having the terms most correlated with each particular query keyword already identified, one more operation is necessary, namely calculating the correlation of every output term with the entire query. Two approaches are possible: (1) using a product of the correlation between the term and all keywords in the original expression (step 5a), or (2) simply counting the number of documents in which the proposed term co-occurs with the entire user query (step 5b). We considered the following formulas for Similarity Coefficients [17]:

- *Cosine Similarity*, defined as:

$$CS = \frac{DF_{x,y}}{\sqrt{DF_x \cdot DF_y}} \tag{2}$$

- *Mutual Information*, defined as:

$$MI = log \frac{N \cdot DF_{x,y}}{DF_x \cdot DF_y} \tag{3}$$

- *Likelihood Ratio*, defined in the paragraphs below.

DF_x is the Document Frequency of term x, and $DF_{x,y}$ is the number of documents containing both x and y. To further increase the quality of the generated scores we limited the latter indicator to co-occurrences within a window of W terms. We set W to be the same as the maximum amount of expansion keywords desired.

Dunning's Likelihood Ratio λ [9] is a co-occurrence based metric similar to χ^2. It starts by attempting to reject the null hypothesis, according to which two terms A and B would appear in text independently from each other. This means that $P(A\ B) = P(A \neg B) = P(A)$, where $P(A \neg B)$ is the probability that term A is *not* followed by term B. Consequently, the test for independence of A and B can be performed by looking if the distribution of A given that B is present is the same as the distribution of A given that B is not present. Of course, in reality we know these terms are not independent in text, and we only use the statistical metrics to highlight terms which are frequently appearing together. We compare the two binomial processes by using likelihood ratios of their associated hypotheses. First, let us define the likelihood ratio for one hypothesis:

$$\lambda = \frac{max_{\omega \in \Omega_0}\ H(\omega; k)}{max_{\omega \in \Omega}\ H(\omega; k)} \tag{4}$$

where ω is a point in the parameter space Ω, Ω_0 is the particular hypothesis being tested, and k is a point in the space of observations K. If we assume that two binomial distributions have the same underlying parameter, i.e., $\{(p_1, p_2) \mid p_1 = p_2\}$, we can write:

$$\lambda = \frac{max_p\ H(p, p; k_1, k_2, n_1, n_2)}{max_{p_1,p_2}\ H(p_1, p_2; k_1, k_2, n_1, n_2)} \tag{5}$$

where $H(p_1, p_2; k_1, k_2, n_1, n_2) = p_1^{k_1} \cdot (1-p_1)^{(n_1-k_1)} \cdot \binom{n_1}{k_1} \cdot p_2^{k_2} \cdot (1-p_2)^{(n_2-k_2)} \cdot \binom{n_2}{k_2}$. Since the maxima are obtained with $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$, and $p = \frac{k_1+k_2}{n_1+n_2}$, we have:

$$\lambda = \frac{max_p\ L(p, k_1, n_1) L(p, k_2, n_2)}{max_{p_1,p_2}\ L(p_1, k_1, n_1) L(p_2, k_2, n_2)} \tag{6}$$

where $L(p, k, n) = p^k \cdot (1-p)^{n-k}$. Taking the logarithm of the likelihood, we obtain:

$$-2 \cdot log\ \lambda = \quad 2 \cdot [log\ L(p_1, k_1, n_1) + log\ L(p_2, k_2, n_2) - log\ L(p, k_1, n_1) - log\ L(p, k_2, n_2)]$$

where $log\ L(p, k, n) = k \cdot log\ p + (n-k) \cdot log(1-p)$. Finally, if we write $O_{11} = P(A\ B)$, $O_{12} = P(\neg A\ B)$, $O_{21} = P(A\ \neg B)$, and $O_{22} = P(\neg A\ \neg B)$, then the co-occurrence likelihood of terms A and B becomes:

$$-2 \cdot log\ \lambda = \quad 2 \cdot [O_{11} \cdot log\ p_1 + O_{12} \cdot log\ (1-p_1) + O_{21} \cdot log\ p_2 + O_{22} \cdot log\ (1-p_2) - (O_{11} + O_{21}) \cdot log\ p - (O_{12} + O_{22}) \cdot log\ (1-p)]$$

where $p_1 = \frac{k_1}{n_1} = \frac{O_{11}}{O_{11}+O_{12}}$, $p_2 = \frac{k_2}{n_2} = \frac{O_{21}}{O_{21}+O_{22}}$, and $p = \frac{k_1+k_2}{n_1+n_2}$

Thesaurus Based Expansion. Large scale thesauri encapsulate global knowledge about term relationships. Thus, we first identify the set of terms closely related to each query keyword, and then we calculate the Desktop co-occurrence level of each of these possible expansion terms with the entire initial search request. In the end, those suggestions with the highest frequencies are kept. The algorithm is as follows:

Algorithm 3.1.2.2. Filtered thesaurus based query expansion.

1: **For** each keyword k of an input query Q:
2: **Select** the following sets of related terms using WordNet:
2a: Syn: All Synonyms
2b: Sub: All sub-concepts residing one level below k
2c: Super: All super-concepts residing one level above k
3: **For** each set S_i of the above mentioned sets:
4: **For** each term t of S_i:
5: **Search** the PIR with $(Q|t)$, i.e.,
the original query, as expanded with t
6: **Let** H be the number of hits of the above search
(i.e., the co-occurence level of t with Q)
7: **Return** Top-K terms as ordered by their H values.

We observe three types of term relationships (steps 2a-2c): (1) synonyms, (2) sub-concepts, namely hyponyms (i.e., sub-classes) and meronyms (i.e., sub-parts), and (3) super-concepts, namely hypernyms (i.e., super-classes) and holonyms (i.e., super-parts). As they represent quite different types of association, we investigated them separately. We limited the output expansion set (step 7) to contain only terms appearing at least T times on the Desktop, in order to avoid noisy suggestions, with $T = \min(\frac{N}{\text{DocsPerTopic}}, \text{MinDocs})$. We set $DocsPerTopic = 2,500$, and $MinDocs = 5$, the latter one coping with the case of small PIRs.

3.2 Experiments

3.2.1 Experimental Setup

We evaluated our algorithms with 18 subjects (Ph.D. and Post-Doc. students in different areas of computer science and education). First, they installed our Lucene based search engine[3] and

[3]Clearly, if one had already installed a Desktop search application, then this overhead would not be present.

indexed all their locally stored content: Files within user selected paths, Emails, and Web Cache. Without loss of generality, we focused the experiments on single-user machines. Then, they chose 4 queries related to their everyday activities, as follows:

- One very frequent AltaVista query, as extracted from the top 2% queries most issued to the search engine within a 7.2 million entries log from October 2001. In order to connect such a query to each user's interests, we added an off-line preprocessing phase: We generated the most frequent search requests and then randomly selected a query with at least 10 hits on each subject's Desktop. To further ensure a real life scenario, users were allowed to reject the proposed query and ask for a new one, if they considered it totally outside their interest areas.
- One randomly selected log query, filtered using the same procedure as above.
- One self-selected specific query, which they thought to have only one meaning.
- One self-selected ambiguous query, which they thought to have at least three meanings.

The average query lengths were 2.0 and 2.3 terms for the log queries, as well as 2.9 and 1.8 for the self-selected ones. Even though our algorithms are mainly intended to enhance search when using ambiguous query keywords, we chose to investigate their performance on a wide span of query types, in order to see how they perform in all situations. The log queries evaluate real life requests, in contrast to the self-selected ones, which target rather the identification of top and bottom performances. Note that the former ones were somewhat farther away from each subject's interest, thus being also more difficult to personalize on. To gain an insight into the relationship between each query type and user interests, we asked each person to rate the query itself with a score of 1 to 5, having the following interpretations: (1) never heard of it, (2) do not know it, but heard of it, (3) know it partially, (4) know it well, (5) major interest. The obtained grades were 3.11 for the top log queries, 3.72 for the randomly selected ones, 4.45 for the self-selected specific ones, and 4.39 for the self-selected ambiguous ones.

For each query, we collected the Top-5 URLs generated by 20 versions of the algorithms[4] presented in Section 3.1. These results were then shuffled into one set containing usually between 70 and 90 URLs. Thus, each subject had to assess about 325 documents for all four queries, being neither aware of the algorithm, nor of the ranking of each assessed URL. Overall, 72 queries were issued and over 6,000 URLs were evaluated during the experiment. For each of these URLs, the testers had to give a rating ranging from 0 to 2, dividing the relevant results in two categories, (1) relevant and (2) highly relevant. Finally, the quality of each ranking was assessed using the normalized version of Discounted Cumulative Gain (DCG) [15]. DCG is a rich measure, as it gives more weight to highly ranked documents, while also incorporating different relevance levels by giving them different gain values:

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

We used G(i) = 1 for relevant results, and G(i) = 2 for highly relevant ones. As queries having more relevant output documents will have a higher DCG, we also normalized its value to a score between 0 (the worst possible DCG given the ratings) and 1 (the best possible DCG given the ratings) to facilitate averaging over queries. All results were tested for statistical significance using T-tests.

[4]Note that all Desktop level parts of our algorithms were performed with Lucene using its predefined searching and ranking functions.

Algorithmic specific aspects. The main parameter of our algorithms is the number of generated expansion keywords. For this experiment we set it to 4 terms for all techniques, leaving an analysis at this level for a subsequent investigation. In order to optimize the run-time computation speed, we chose to limit the number of output keywords per Desktop document to the number of expansion keywords desired (i.e., four). For all algorithms we also investigated bigger limitations. This allowed us to observe that the Lexical Compounds method would perform better if only at most one compound per document were selected. We therefore chose to experiment with this new approach as well. For all other techniques, considering less than four terms per document did not seem to consistently yield any additional qualitative gain. We labeled the algorithms we evaluated as follows:

0. **Google**: The actual Google query output, as returned by the Google API;
1. **TF, DF**: Term and Document Frequency;
2. **LC, LC[O]**: Regular and Optimized (by considering only one top compound per document) Lexical Compounds;
3. **SS**: Sentence Selection;
4. **TC[CS], TC[MI], TC[LR]**: Term Co-occurrence Statistics using respectively Cosine Similarity, Mutual Information, and Likelihood Ratio as similarity coefficients;
5. **WN[SYN], WN[SUB], WN[SUP]**: WordNet based expansion with synonyms, sub-concepts, and super-concepts, respectively.

Except for the thesaurus based expansion, in all cases we also investigated the performance of our algorithms when exploiting only the Web browser cache to represent user's personal information. This is motivated by the fact that other personal documents such as for example emails are known to have a somewhat different language than that residing on the world wide Web [34]. However, as this approach performed visibly poorer than using the entire Desktop data, we omitted it from the subsequent analysis.

3.2.2 Results

Log Queries. We evaluated all variants of our algorithms using NDCG. For log queries, the best performance was achieved with TF, LC[O], and TC[LR]. The improvements they brought were up to 5.2% for top queries ($p = 0.14$) and 13.8% for randomly selected queries ($p = 0.01$, statistically significant), both obtained with LC[O]. A summary of all results is depicted in Table 1.

Both TF and LC[O] yielded very good results, indicating that simple keyword and expression oriented approaches might be sufficient for the Desktop based query expansion task. LC[O] was much better than LC, ameliorating its quality with up to 25.8% in the case of randomly selected log queries, improvement which was also significant with $p = 0.04$. Thus, a selection of compounds spanning over several Desktop documents is more informative about user's interests than the general approach, in which there is no restriction on the number of compounds produced from every personal item.

The more complex Desktop oriented approaches, namely sentence selection and all term co-occurrence based algorithms, showed a rather average performance, with no visible improvements, except for TC[LR]. Also, the thesaurus based expansion usually produced very few suggestions, possibly because of the many technical queries employed by our subjects. We observed however that expanding with sub-concepts is very good for everyday life terms (e.g., "car"), whereas the use of super-concepts is valuable for compounds having at least one term with low technicality (e.g., "document clustering"). As expected, the synonym based expansion performed generally well, though in some very

Algorithm	NDCG Top	Signific. vs. Google	NDCG Random	Signific. vs. Google
Google	0.42	-	0.40	-
TF	**0.43**	$\mathbf{p = 0.32}$	**0.43**	$\mathbf{p = 0.04}$
DF	0.17	-	0.23	-
LC	0.39	-	0.36	-
LC[O]	**0.44**	$\mathbf{p = 0.14}$	**0.45**	$\mathbf{p = 0.01}$
SS	0.33	-	0.36	-
TC[CS]	0.37	-	0.35	-
TC[MI]	0.40	-	0.36	-
TC[LR]	0.41	-	**0.42**	$\mathbf{p = 0.06}$
WN[SYN]	0.42	-	0.38	-
WN[SUB]	0.28	-	0.33	-
WN[SUP]	0.26	-	0.26	-

Table 1: Normalized Discounted Cumulative Gain at the first 5 results when searching for top (left) and random (right) log queries.

Algorithm	NDCG Clear	Signific. vs. Google	NDCG Ambiguous	Signific. vs. Google
Google	0.71	-	0.39	-
TF	0.66	-	**0.52**	$\mathbf{p \ll 0.01}$
DF	0.37	-	0.31	-
LC	0.65	-	**0.54**	$\mathbf{p \ll 0.01}$
LC[O]	0.69	-	**0.59**	$\mathbf{p \ll 0.01}$
SS	0.56	-	**0.52**	$\mathbf{p \ll 0.01}$
TC[CS]	0.60	-	**0.50**	$\mathbf{p = 0.01}$
TC[MI]	0.60	-	**0.47**	$\mathbf{p = 0.02}$
TC[LR]	0.56	-	**0.47**	$\mathbf{p = 0.03}$
WN[SYN]	0.70	-	0.36	-
WN[SUB]	0.46	-	0.32	-
WN[SUP]	0.51	-	0.29	-

Table 2: Normalized Discounted Cumulative Gain at the first 5 results when searching for user selected clear (left) and ambiguous (right) queries.

technical cases it yielded rather general suggestions. Finally, we noticed Google to be very optimized for some top frequent queries. However, even within this harder scenario, some of our personalization algorithms produced statistically significant improvements over regular search (i.e., TF and LC[O]).

Self-selected Queries. The NDCG values obtained with self-selected queries are depicted in Table 2. While our algorithms did not enhance Google for the clear search tasks, they did produce strong improvements of up to 52.9% (which were of course also highly significant with $p \ll 0.01$) when utilized with ambiguous queries. In fact, almost all our algorithms resulted in statistically significant improvements over Google for this query type.

In general, the relative differences between our algorithms were similar to those observed for the log based queries. As in the previous analysis, the simple Desktop based Term Frequency and Lexical Compounds metrics performed best. Nevertheless, a very good outcome was also obtained for Desktop based sentence selection and all term co-occurrence metrics. There were no visible differences between the behavior of the three different approaches to co-occurrence calculation. Finally, for the case of clear queries, we noticed that fewer expansion terms than 4 might be less noisy and thus helpful in bringing further improvements. We thus pursued this idea with the adaptive algorithms presented in the next section.

4. INTRODUCING ADAPTIVITY

In the previous section we have investigated the behavior of each technique when adding a fixed number of keywords to the user query. However, an optimal personalized query expansion algorithm should automatically adapt itself to various aspects of each query, as well as to the particularities of the person using it. In this section we discuss the factors influencing the behavior of our expansion algorithms, which might be used as input for the adaptivity process. Then, in the second part we present some initial experiments with one of them, namely query clarity.

4.1 Adaptivity Factors

Several indicators could assist the algorithm to automatically tune the number of expansion terms. We start by discussing adaptation by analyzing the query clarity level. Then, we briefly introduce an approach to model the generic query formulation process in order to tailor the search algorithm automatically, and discuss some other possible factors that might be of use for this task.

Query Clarity. The interest for analyzing query difficulty has increased only recently, and there are not many papers addressing this topic. Yet it has been long known that query disambiguation has a high potential of improving retrieval effectiveness for low recall searches with very short queries [20], which is exactly our targeted scenario. Also, the success of IR systems clearly varies across different topics. We thus propose to use an estimate number expressing the calculated level of query clarity in order to automatically tweak the amount of personalization fed into the algorithm. The following metrics are available:

- *The Query Length* is expressed simply by the number of words in the user query. The solution is rather inefficient, as reported by He and Ounis [14].
- *The Query Scope* relates to the IDF of the entire query, as in:

$$C_1 = log(\frac{\#DocumentsInCollection}{\#Hits(Query)}) \quad (7)$$

  This metric performs well when used with document collections covering a single topic, but poor otherwise [7, 14].
- *The Query Clarity* [7] seems to be the best, as well as the most applied technique so far. It measures the divergence between the language model associated to the user query and the language model associated to the collection. In a simplified version (i.e., without smoothing over the terms which are not present in the query), it can be expressed as follows:

$$C_2 = \sum_{w \in Query} P_{ml}(w|Query) \cdot log\frac{P_{ml}(w|Query)}{P_{coll}(w)} \quad (8)$$

  where $P_{ml}(w|Query)$ is the probability of the word w within the submitted query, and $P_{coll}(w)$ is the probability of w within the entire collection of documents.

Other solutions exist, but we think they are too computationally expensive for the huge amount of data that needs to be processed within Web applications. We thus decided to investigate only C_1 and C_2. First, we analyzed their performance over a large set of queries and split their clarity predictions in three categories:

- Small Scope / Clear Query: $C_1 \in [0, 12], C_2 \in [4, \infty)$.
- Medium Scope / Semi-Ambiguous Query: $C_1 \in [12, 17), C_2 \in [2.5, 4)$.
- Large Scope / Ambiguous Query: $C_1 \in [17, \infty), C_2 \in [0, 2.5]$.

In order to limit the amount of experiments, we analyzed only the results produced when employing C_1 for the PIR and C_2 for the Web. As algorithmic basis we used LC[O], i.e., optimized lexical compounds, which was clearly the winning method in the previous analysis. As manual investigation showed it to slightly overfit the expansion terms for clear queries, we utilized a substitute for this particular case. Two candidates were considered: (1) TF, i.e., the second best approach, and (2) WN[SYN], as we observed that its first and second expansion terms were often very good.

Desktop Scope	Web Clarity	No. of Terms	Algorithm
Large	Ambiguous	4	LC[O]
Large	Semi-Ambig.	3	LC[O]
Large	Clear	2	LC[O]
Medium	Ambiguous	3	LC[O]
Medium	Semi-Ambig.	2	LC[O]
Medium	Clear	1	TF / WN[SYN]
Small	Ambiguous	2	TF / WN[SYN]
Small	Semi-Ambig.	1	TF / WN[SYN]
Small	Clear	0	-

Table 3: Adaptive Personalized Query Expansion.

Given the algorithms and clarity measures, we implemented the adaptivity procedure by tailoring the amount of expansion terms added to the original query, as a function of its ambiguity in the Web, as well as within user's PIR. Note that the ambiguity level is related to the number of documents covering a certain query. Thus, to some extent, it has different meanings on the Web and within PIRs. While a query deemed ambiguous on a large collection such as the Web will very likely indeed have a large number of meanings, this may not be the case for the Desktop. Take for example the query "PageRank". If the user is a link analysis expert, many of her documents might match this term, and thus the query would be classified as ambiguous. However, when analyzed against the Web, this is definitely a clear query. Consequently, we employed more additional terms, when the query was more ambiguous in the Web, but also on the Desktop. Put another way, queries deemed clear on the Desktop were inherently not well covered within user's PIR, and thus had fewer keywords appended to them. The number of expansion terms we utilized for each combination of scope and clarity levels is depicted in Table 3.

Query Formulation Process. Interactive query expansion has a high potential for enhancing search [29]. We believe that modeling its underlying process would be very helpful in producing qualitative adaptive Web search algorithms. For example, when the user is adding a new term to her previously issued query, she is basically reformulating her original request. Thus, the newly added terms are more likely to convey information about her search goals. For a general, non personalized retrieval engine, this could correspond to giving more weight to these new keywords. Within our personalized scenario, the generated expansions can similarly be biased towards these terms. Nevertheless, more investigations are necessary in order to solve the challenges posed by this approach.

Other Features. The idea of adapting the retrieval process to various aspects of the query, of the user itself, and even of the employed algorithm has received only little attention in the literature. Only some approaches have been investigated, usually indirectly. There exist studies of query behaviors at different times of day, or of the topics spanned by the queries of various classes of users, etc. However, they generally do not discuss how these features can be actually incorporated in the search process itself and they have almost never been related to the task of Web personalization.

4.2 Experiments

We used exactly the same experimental setup as for our previous analysis, with two log-based queries and two self-selected ones (all different from before, in order to make sure there is no bias on the new approaches), evaluated with NDCG over the Top-5 results output by each algorithm. The newly proposed adaptive personalized query expansion algorithms are denoted as A[LCO/TF] for the approach using TF with the clear Desktop queries, and as A[LCO/WN] when WN[SYN] was utilized instead of TF.

The overall results were at least similar, or better than Google for all kinds of log queries (see Table 4). For top frequent queries,

Algorithm	NDCG Top	Signific. vs. Google	NDCG Random	Signific. vs. Google
Google	0.51	-	0.45	-
TF	**0.51**	-	**0.48**	$\mathbf{p = 0.04}$
LC[O]	**0.53**	$\mathbf{p = 0.09}$	**0.52**	$\mathbf{p < 0.01}$
WN[SYN]	**0.51**	-	**0.45**	-
A[LCO/TF]	**0.56**	$\mathbf{p < 0.01}$	**0.49**	$\mathbf{p = 0.04}$
A[LCO/WN]	**0.55**	$\mathbf{p = 0.01}$	0.44	-

Table 4: Normalized Discounted Cumulative Gain at the first 5 results when using our *adaptive* personalized search algorithms on top (left) and random (right) log queries.

Algorithm	NDCG Clear	Signific. vs. Google	NDCG Ambiguous	Signific. vs. Google
Google	0.81	-	0.46	-
TF	0.76	-	**0.54**	$\mathbf{p = 0.03}$
LC[O]	0.77	-	**0.59**	$\mathbf{p \ll 0.01}$
WN[SYN]	0.79	-	0.44	-
A[LCO/TF]	**0.81**	-	**0.64**	$\mathbf{p \ll 0.01}$
A[LCO/WN]	**0.81**	-	**0.63**	$\mathbf{p \ll 0.01}$

Table 5: Normalized Discounted Cumulative Gain at the first 5 results when using our *adaptive* personalized search algorithms on user selected clear (left) and ambiguous (right) queries.

both adaptive algorithms, A[LCO/TF] and A[LCO/WN], improve with 10.8% and 7.9% respectively, both differences being also statistically significant with $p \leq 0.01$. They also achieve an improvement of up to 6.62% over the best performing static algorithm, LC[O] ($p = 0.07$). For randomly selected queries, even though A[LCO/TF] yields significantly better results than Google ($p = 0.04$), both adaptive approaches fall behind the static algorithms. The major reason seems to be the imperfect selection of the number of expansion terms, as a function of query clarity. Thus, more experiments are needed in order to determine the optimal number of generated expansion keywords, as a function of the query ambiguity level.

The analysis of the self-selected queries shows that adaptivity can bring even further improvements into Web search personalization (see Table 5). For ambiguous queries, the scores given to Google search are enhanced by 40.6% through A[LCO/TF] and by 35.2% through A[LCO/WN], both strongly significant with $p \ll 0.01$. Adaptivity also brings another 8.9% improvement over the static personalization of LC[O] ($p = 0.05$). Even for clear queries, the newly proposed flexible algorithms perform slightly better, improving with 0.4% and 1.0% respectively.

All results are depicted graphically in Figure 1. We notice that A[LCO/TF] is the overall best algorithm, performing better than Google for all types of queries, either extracted from the search engine log, or self-selected. The experiments presented in this section confirm clearly that adaptivity is a necessary further step to take in Web search personalization.

5. CONCLUSIONS AND FURTHER WORK

In this paper we proposed to expand Web search queries by exploiting the user's Personal Information Repository in order to automatically extract additional keywords related both to the query itself and to user's interests, personalizing the search output. In this context, the paper includes the following contributions:

- We proposed five techniques for determining expansion terms from personal documents. Each of them produces additional query keywords by analyzing user's Desktop at increasing granularity levels, ranging from term and expression level analysis up to global co-occurrence statistics and external thesauri.

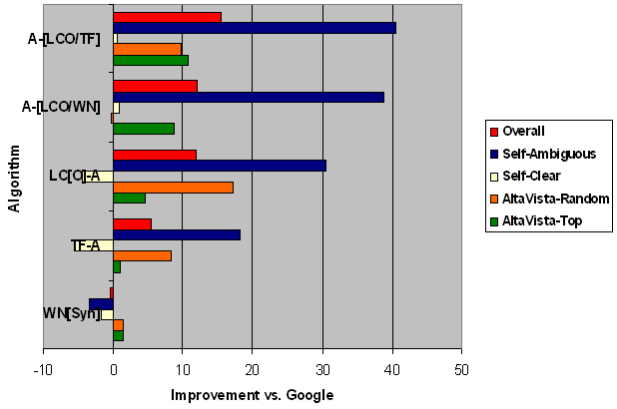

Figure 1: Relative NDCG gain (in %) for each algorithm overall, as well as separated per query category.

- We provided a thorough empirical analysis of several variants of our approaches, under four different scenarios. We showed some of these approaches to perform very well, producing NDCG improvements of up to 51.28%.
- We moved this personalized search framework further and proposed to make the expansion process adaptive to features of each query, a strong focus being put on its clarity level.
- Within a separate set of experiments, we showed our adaptive algorithms to provide an additional improvement of 8.47% over the previously identified best approach.

We are currently performing investigations on the dependency between various query features and the optimal number of expansion terms. We are also analyzing other types of approaches to identify query expansion suggestions, such as applying Latent Semantic Analysis on the Desktop data. Finally, we are designing a set of more complex combinations of these metrics in order to provide enhanced adaptivity to our algorithms.

6. ACKNOWLEDGEMENTS

We thank Ricardo Baeza-Yates, Vassilis Plachouras, Carlos Castillo and Vanessa Murdock from Yahoo! for the interesting discussions about the experimental setup and the algorithms we presented. We are grateful to Fabrizio Silvestri from CNR and to Ronny Lempel from IBM for providing us the AltaVista query log. Finally, we thank our colleagues from L3S for participating in the time consuming experiments we performed, as well as to the European Commission for the funding support (project Nepomuk, 6th Framework Programme, IST contract no. 027705).

7. REFERENCES

[1] J. Allan and H. Raghavan. Using part-of-speech patterns to reduce query ambiguity. In *Proc. of the 25th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, 2002.

[2] P. G. Anick and S. Tipirneni. The paraphrase search assistant: Terminological feedback for iterative information seeking. In *Proc. of the 22nd Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1999.

[3] D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic query wefinement using lexical affinities with maximal information gain. In *Proc. of the 25th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, pages 283–290, 2002.

[4] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM TOIS*, 19(1):1–27, 2001.

[5] C.-H. Chang and C.-C. Hsu. Integrating query expansion and conceptual relevance feedback for personalized web information retrieval. In *Proc. of the 7th Intl. Conf. on World Wide Web*, 1998.

[6] P. A. Chirita, C. Firan, and W. Nejdl. Summarizing local context to personalize global web search. In *Proc. of the 15th Intl. CIKM Conf. on Information and Knowledge Management*, 2006.

[7] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proc. of the 25th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, 2002.

[8] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *Proc. of the 11th Intl. Conf. on World Wide Web*, 2002.

[9] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19:61–74, 1993.

[10] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285, 1969.

[11] E. N. Efthimiadis. User choices: A new yardstick for the evaluation of ranking algorithms for interactive query expansion. *Information Processing and Management*, 31(4):605–620, 1995.

[12] D. Fogaras and B. Racz. Scaling link based similarity search. In *Proc. of the 14th Intl. World Wide Web Conf.*, 2005.

[13] T. Haveliwala. Topic-sensitive pagerank. In *Proc. of the 11th Intl. World Wide Web Conf., Honolulu, Hawaii*, May 2002.

[14] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *Proc. of the 11th Intl. SPIRE Conf. on String Processing and Information Retrieval*, 2004.

[15] K. Järvelin and J. Keklinen. Ir evaluation methods for retrieving highly relevant documents. In *Proc. of the 23th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, 2000.

[16] G. Jeh and J. Widom. Scaling personalized web search. In *Proc. of the 12th Intl. World Wide Web Conference*, 2003.

[17] M.-C. Kim and K.-S. Choi. A comparison of collocation-based similarity measures in query expansion. *Inf. Proc. and Mgmt.*, 35(1):19–30, 1999.

[18] S.-B. Kim, H.-C. Seo, and H.-C. Rim. Information retrieval using word senses: root sense tagging approach. In *Proc. of the 27th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, 2004.

[19] R. Kraft and J. Zien. Mining anchor text for query refinement. In *Proc. of the 13th Intl. Conf. on World Wide Web*, 2004.

[20] R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst.*, 10(2), 1992.

[21] A. M. Lam-Adesina and G. J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In *Proc. of the 24th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2001.

[22] S. Liu, F. Liu, C. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proc. of the 27th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, 2004.

[23] G. Miller. Wordnet: An electronic lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

[24] L. Nie, B. Davison, and X. Qi. Topical link analysis for web search. In *Proc. of the 29th Intl. ACM SIGIR Conf. on Res. and Development in Inf. Retr.*, 2006.

[25] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Univ., 1998.

[26] F. Qiu and J. Cho. Automatic indentification of user interest for personalized search. In *Proc. of the 15th Intl. WWW Conf.*, 2006.

[27] Y. Qiu and H.-P. Frei. Concept based query expansion. In *Proc. of the 16th Intl. ACM SIGIR Conf. on Research and Development in Inf. Retr.*, 1993.

[28] J. Rocchio. Relevance feedback in information retrieval. *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.

[29] I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In *Proc. of the 26th Intl. ACM SIGIR Conf.*, 2003.

[30] T. Sarlos, A. A. Benczur, K. Csalogany, D. Fogaras, and B. Racz. To randomize or not to randomize: Space optimal summaries for hyperlink analysis. In *Proc. of the 15th Intl. WWW Conf.*, 2006.

[31] C. Shah and W. B. Croft. Evaluating high accuracy retrieval techniques. In *Proc. of the 27th Intl. ACM SIGIR Conf. on Research and development in information retrieval*, pages 2–9, 2004.

[32] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. of the 13th Intl. World Wide Web Conf.*, 2004.

[33] D. Sullivan. The older you are, the more you want personalized search, 2004. http://searchenginewatch.com/searchday/article.php/3385131.

[34] J. Teevan, S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2005.

[35] E. Volokh. Personalization and privacy. *Commun. ACM*, 43(8), 2000.

[36] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proc. of the 17th Intl. ACM SIGIR Conf. on Res. and development in Inf. Retr.*, 1994.

[37] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proc. of the 19th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1996.

[38] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proc. of the 12th Intl. Conf. on World Wide Web*, 2003.