Personalized Faceted Query Expansion

Haward Jie
Computer Science Department
University of California Santa Cruz
haward@cse.ucsc.edu

Yi Zhang
School of Engineering
University of California Santa Cruz
yiz@soe.ucsc.edu

ABSTRACT

Search engines provide advanced functionalities to let the user put constraints on document facets. These functionalities can be helpful if they are applied properly, however the average user often does not take advantage of them**.**

Instead of waiting for a user to issue a complex query, we propose to expand automatically from a free text query into a semi-structured query using pseudo-relevance feedback. To tailor the expanded query to the user's information need, a large amount of information on the user's local computer or personal device is used to generate pseudo-relevant documents for query expansion. We developed a personalized search engine called **"**YoooYooo**"** that is based on this concept. The search engine recommends the expanded query to the user as facet constraints and allows the user to take the control of the personalized search. Preliminary experimental results showed a statistically significant improvement over Google and Yahoo.

1. INTRODUCTION

Search engines provide advanced functionalities to let the user place constraints on document facets, such as limiting the scope of the search to specific areas of the site or limiting the results to a specific language[1][2][3]. Ideally advanced search functionalities can be useful if applied properly. For example, a user wanting to download Machine Learning papers can use a structured query**,** such as "Machine Learning filetype:pdf" to narrow the search results to pdf files, or a user can use a structured query**,** such as "Machine Learning filetype:ppt" to limit the search results to powerpoint (ppt) slides. However, the average user is not good at issuing complex queries, or he/she is often unwilling to make the effort to do so. Thus most users do not use advanced search functionalities, and the mean query length is only about 2.4 words, according to a study based on 60,000,000 searches [5].

Instead of relying on the user to make the effort to learn complex query languages and to create a structured query manually, a "smart" search engine can actively recommend advanced queries to the user. The large amount of information available on a user's local computer or personal device provides an opportunity for a search engine to learn about the user and to tailor the search results to the his/her information needs [6][8].

To take advantage of such advanced functionalities and rich information about the user, we propose that a search engine automatically generate and recommend personalized semi-structured queries. We suggest using pseudo-relevance feedback that is generated from the local documents the user has previously read to create the queries that have been automatically defined by facets of documents. In this paper, a simple translation model for semi-structured query expansion is described, a personalized search engine is developed using this concept, and preliminary experimental results are reported.

2. PERSONALIZED FACET QUERY EXPANSION

2.1 A Translation Model for Facet Query Expansion

Let Q_0 be the initial query issued by the user. Each document has a set of meta-data fields, each of which corresponds to a facet. We convert each meta-data to a facet token. For example, for the site facet, the facet token can be *site: ucsc.edu*.

Let $d_a = (d, d_f) = (w_1, w_2, ..., w_K, t_1, t_2, ..., t_M)$ be a document vector that represents the document augmented with facet tokens where $d = (w_1, w_2, ..., w_K)$ is the original document vector indexed by *K* normal keywords, while $d_f = (t_1, t_2, ..., t_M)$ is the vector indexed by *M* facet types. For example, if {language, file type, …} is the set of possible facet types, then the first facet dimension t_1 reveals the language of the document; the second facet dimension tells the file type of the document, and so on.

Given the initial user query Q_0, the next step is to expand it into a new query $Q = \{Q_0, Q_f\} = \{Q_0, t_1, t_2, t_3, ..., t_M\}$, where $Q_f = \{t_1, t_2, t_3, ..., t_M\}$ is a set of query terms (facet tokens) that have been added. It can also be viewed as the scoping constraints for relevant documents as defined by the corresponding facets.

There is a large amount of literature on query expansion**;** however**,** how to create a semi-structured query from free text query has rarely been studied. Motivated by [4], we take the view that query expansion is a form of translation from one language (the original query language, usually free text) to another language (the added query, usually a sequence of facets tokens)**.** The optimal translation is:

$$Q_f = \arg\max_{Q_k} P(Q_f = \{t_1, t_2, t_3, ..., t_M\} \mid Q_0) \qquad (1)$$

We assume the original query Q_o is translated into a new Q_f by making several independent translations of the query Q_o to a single facet token t_i in the following manner (Figure 1). For the *i*th facet type, the statistical translator first randomly chooses a document d_j according to $P(d|Q_o)$, then it randomly chooses a facet token t_i according to $P_i(t_i|d_j)$. Thus the probability generating Q_f from the original query Q_o is:

$$P(Q_f = \{t_1, t_2, ..., t_M\} \mid Q_0) = \prod_{i=1}^{M} P(t_i \mid Q_0) = \prod_{i=1}^{M} \sum_{d_j} P(t_i \mid d_j) P(d_j \mid Q_0) \qquad (2)$$

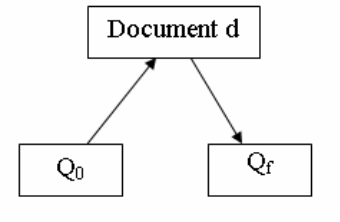

Figure 1 Translation model for query expansion

2.2 Estimating Model Parameters using User History

We need to estimate two sets of model parameters: $P(d_j|Q)$, the probability of picking document d_j given the original query, and $P_i(t_i|d_j)$, the probability of generating facet value t_i for each facet dimension *i*.

If there are *N* relevant documents, ideally, $P(d_j|Q)=1/N$ if d_j is relevant to query *Q*, otherwise $P(d_j|Q)=0$. However, we do not know which documents are relevant to a newly issued user query. Instead, we ran the original query *Qo* against the user's local data to retrieve the top *N* documents and assumed they are relevant. Thus $P(d_j|Q_o)=1/N$ for documents retrieved and 0 for other documents.

Estimating $P(t_i|d_j)$ is straightforward: $P(t_i|d_j)$ =1 if t_i is a *i*th dimension facet token associated with document *j*, otherwise $P(t_i|d_j)=0$. For example, if document d_j is a pdf file written in English, $P(t_1=<language:English>|d_j)=1$, $P(t_1=<language:German>|d_j)=0$, $P(t_2=<filetype:pdf>|d_j)=1$, $P(t_2=<filetype:ppt>|d_j)=0$.

Thus Equation 2 becomes a simple form, and the likelihood of a new query Q_f can be estimated using:

$$P(Q_f = \{t_1, t_2, ..., t_M\} \mid Q_0) = \prod_{i=1}^{M} \frac{\#(t_i)}{N}$$

where $\#(t_i)$ is the document frequency of facet token t_i over pseudo-relevant documents. Since the whole translation process generates each facet value independent of the value of other dimensions, the likelihood of each facet token t_i is $P(t_i \mid Q_0)$. For example, if $1 \geq P(t_1 = < filetype : pdf > \mid Q_0) = 0.9$, $P(t_2 = < site : u\,\mathrm{csc}.edu > \mid Q_0) = 0$, and $1 \geq P(t_3 = < age : 12 > \mid Q_0) = 0.5$, it indicates that 90% of the top ranking local documents are pdf files, that none of those documents were downloaded from the *ucsc.edu* domain, and that 50% of the documents are older than twelve months.

The whole process simulates a manual query modification carried out by a user. The user formulates a semi-structured query $Q = \{Q_0, Q_f\} = \{Q_0, t_1, t_2, t_3, ..., t_M\}$ by adding the most likely constraints on each facet to narrow down the results after seeing a sample of relevant documents. However, rather than relying on the user to tell the system which documents are relevant, we assume the top *N* documents retrieved from the local machines are relevant documents the user has seen. This is similar to a commonly used approach called pseudo-relevance feedback, which has been found to be highly effective in non-personalized settings [7]. In our work, we have utilized the user's local data, and this enables the system to tailor-make the expanded query to the individual user.

2.3 Recommending Expanded Query to the User

The query expansion model is based on several assumptions that may not be true for some user queries. If the current query is related to the user's long-term information needs and is on a topic the user has previously accessed, there should be some relevant documents in the user's local file system, and the proposed approach is likely to generate a good quality augmented query. If the query is related to a user's new interest, the system may not be able to obtain a good quality augmented query, since the top-ranking documents retrieved from the user's local file system are unlikely to be relevant**.** For that reason, simply issuing the augmented query to a search engine does not always provide better results than the original query. Results of preliminary experiments confirmed this. Instead of doing automatic personalization by returning to the user the results of the expanded query, we designed an interface to help alleviate inaccurate personalization and let the user decide whether to accept a query expansion recommendation or not. When the initial query keywords are ambiguous and have multiple meanings, this approach allows the user to help the search engine focus on specific interests, for example by instructing the search engine to limit the results within a particular site or searching for a particular file type.

3. EXPERIMENTS

3.1 Experimental Design

To evaluate the proposed technique, we developed a search engine called YoooYooo on top of existing search engines. MSN desktop search API is used to retrieve the documents in the user's local computer for query expansion. When a user issues a query, YoooYooo first returns the results from Google for the original query, together with recommended query expansions for selected facets (Figure 2). The user can narrow the search results by selecting the facets recommended by YoooYooo. Based on the chosen facets, YoooYooo re-submits the expanded query to Google and returns the refined search results to the user (Figure 3). Google's APIs are used in our experiment. Thus the facets are limited to the eight dimensions (image, video, file type, language, date, domain, phrase, safe search) supported by the APIs. We assumed that each facet dimension has a recommended value in the expanded query Q_f. The system can choose not to provide recommendations for a particular facet dimension. Because the current system only supports eight facet dimensions, we let all dimensions with non-zero probability show up on the recommended list by setting a small positive threshold over $P(t_i \mid Q_0)$ for each dimension**.**

We carried out some preliminary experiments to evaluate the quality of the personalized search system by comparing it with two of the most popular search engines: Yahoo and Google. The precision of the top *K* retrieved documents is used as the evaluation measure, where

$$precision = \frac{\text{the number of relevant documents retrieved}}{\text{the total number of documents retrieved}}$$

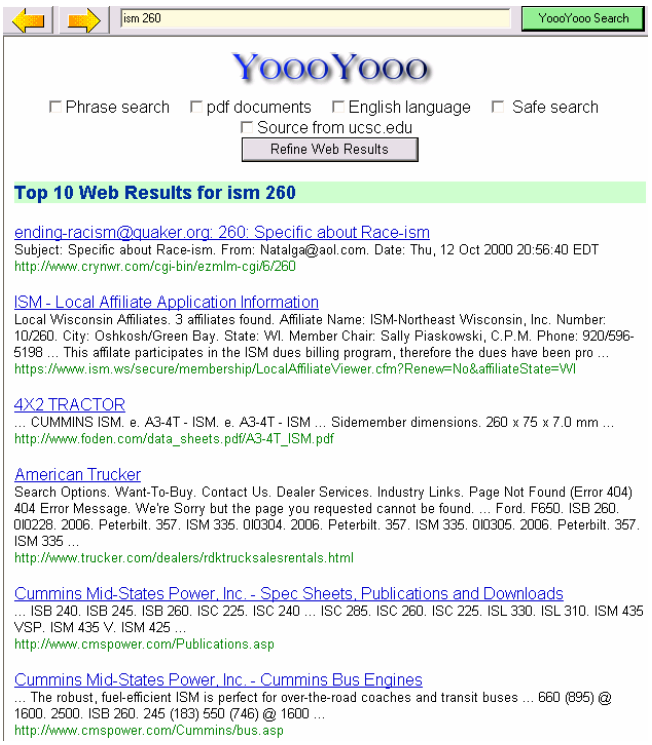

Figure 2: Search Results for Query = *ism 260* before query expansion. The user is a student and the intention of this search is to find the course web site for the course ISM 260.

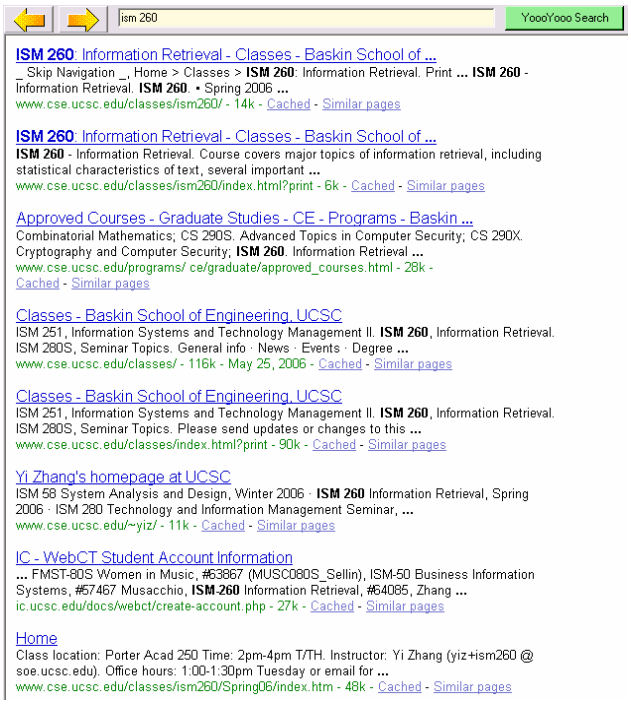

Figure 3: Search results for query = *ism 260* after query expansion. Three facets were selected: Phrase search, English language, and Source from ucsc.edu.

	YoooYooo	Google	Yahoo
P@10	**0.79**	0.71	0.63

Table 1: Comparing the performance of three search engines. P@10 is the average precision at top 10.

	Keywords	YoooYooo	Google	Yahoo
1	active learning multinomial distribution	8/10	8/10	8/10
2	semisupervised clustering learning	10/10	9/10	8/10
3	lcd monitor deal	10/10	10/10	10/10
4	NBA playoff schedule	10/10	10/10	9/10
5	ism 260 hw2	2/2	2/2	0/10
6	active learning support vector machine	10/10	10/10	10/10
7	mission impossible 3 review	10/10	10/10	10/10
8	feature selection information gain	10/10	10/10	10/10
9	memorial day activity in sunnyvale	9/10	9/10	5/10
10	text classifier software	10/10	10/10	10/10

Table 2: The results for user 1. P@10 is reported.

	Keywords	YoooYooo	Google	Yahoo
1	picture processing	10/10	10/10	8/10
2	pattern recognition	10/10	10/10	10/10
3	object representation	10/10	7/10	8/10
4	tracking	3/10	0/10	0/10
5	dynamic model	10/10	7/10	8/10
6	image features	10/10	9/10	8/10
7	scale space	10/10	9/10	5/10
8	subspace	10/10	1/10	2/10
9	non-orthogonal binary space	9/10	8/10	5/10
10	non-photorealistic rendering	10/10	10/10	10/10

Table 3: The results for user 2. P@10 is reported

	Keywords	YoooYooo	Google	Yahoo
1	pink floyd	10/10	10/10	10/10
2	python tutorial	6/10	6/10	7/10
3	santa cruz spca foster	2/10	2/10	2/10
4	cost sensitive boosting	4/10	7/10	6/10
5	tree alignment information retrieval	4/10	2/10	1/10
6	santa cruz live music	5/10	5/10	5/10
7	streaming music recommendation	2/10	2/10	5/10
8	getting rid of ants	6/10	5/10	6/10
9	information retrieval job opportunities	6/10	6/10	1/10
10	linear programming belief propagation	2/10	2/10	2/10

Table 4: The results for user 3. P@10 is reported.

	Initial Query	Expanded Query
1	Object representation	Object representation [filetype:pdf]
2	Semi-supervised clustering learning	Semi-supervised clustering learning [filetype:pdf]
3	Santa Cruz live music	Santa Cruz live music [lr=lang_en]

Table 5: Samples of Initial Query vs. Expanded Query

We set *K=10*, since all three search engines display 10 documents as the search results on the first page, and most of the users only select results from the first page.

There is no publicly available standard data set by which to evaluate the personalized search environment proposed here, so we carried out a small-scale user study. Three subjects from the Computer Science Department or Computer Engineering Department at the University of California Santa Cruz participated in the study as volunteers. Each subject submitted 10 queries that were related to his current interests. The top 10 documents retrieved by each search engine were presented to the subject. Then the subject provided his/her judgment regarding the relevance of each document returned. This information was used to calculate the relevance of the results that had been returned. YoooYooo was running on the user's machine, and the documents evaluated were those returned after receiving the user's feedback over facets.

3.2 Experimental Results

Table 1 compares YoooYooo, Google and Yahoo's average relevance of the top 10 retrieved documents. Further tests of the hypothesis indicated that YoooYooo is statistically significantly better than Google and Yahoo (T test).

The approach described in this paper provides an easy-to-use mechanism by which the user can narrow down search results and improve search quality using selected facets. We hypothesized that this seldom hurts performance, because the system allows the user to control the extent of personalization. To test the hypothesis and better understand how the proposed technique works, each user's queries and the corresponding performance of each search engine are listed in Table 2, Table 3, and Table 4. Samples of original queries and expanded queries are provided in Table 5. Compared to Google, the performance of the proposed technique was the same for 19 queries, better on 10 queries, and less successful on one query.

The users seldom took advantage of the advanced search functionality prior to the study. Seventy percent of the time the users selected one or more facets in order to expand a query. The fact that they chose to use facets so frequently may be the result of several factors: 1) their curiosity about the new interface; 2) their lack of satisfaction about the first set of results; 3) the simplicity of the interface; and 4) the bias caused by the experimental setting.

Only three subjects participated in this study, and only 30 queries were considered. The results suggest that a biased sample of all possible user scenarios were covered in this small-scale user study. Further analysis showed that the users often selected two specific facets (language and files types), and this suggests the existence of independent facet preferences relative to specific queries.

4. FURTHER DISCUSSION AND FUTURE WORK

Instead of passively waiting for the user to make the effort to create a structured query, the YoooYooo search engine can actively provide recommendations to the user by learning from the user's local information. Benefits of doing this would be the following: 1) it would be an easy way for the user to narrow the search results by simply choosing the suggested constraints over the major facets of the documents; 2) it would enable the user to control personalization; and 3) the system would implicitly teach the user about the advanced functionalities provided by the search engine. The results of the preliminary experiments reported in this paper indicated that the proposed approach is promising.

The work described herein is a first step toward creating a personalized structural query expansion. Only eight facets were used in this study due to the limitations imposed by Google. Other document facets could be added in the future. As more facets are added, the system will need to select query-specific facets carefully so that it will only recommend a small set of relevant facets.

The proposed query expansion model can be improved so that the system can recommend better queries to the user. The presently applied expanded query language supports only independent constraints on facets of the document, while a more complex structured or semi-structured query language could be explored in the future. In addition, we can develop different translation models other than the one described in this paper, and this gives us much room to improve the proposed technique. In the present study, we let $P(t_i|d_j) = 1$ if the ith dimension facet meta field of document j is t_i, otherwise $P(t_i|d_j)=0$. However, different values of a facet dimension are not necessarily mutually exclusive. For example, if d_j came from *soe.ucsc.edu*, an alternative translation model could let $P(t_i = \text{"soe.ucsc.edu"} \mid d_j) = 1$, $P(t_i = \text{"ucsc.edu"} \mid d_j) = 1$, and $P(t_i = \text{"edu"} \mid d_j) = 1$.

Although YoooYooo is running on the user's local machine, the proposed technique could also be implemented on the server side**,** if the provided server maintains a sufficient user history.

5. ACKNOWLEDGMENTS

We thank to Zuobing Xu, Philip Zigoris and Feng Tang for helping us with the user evaluations. We also thank Yahoo! for supporting part of the research reported in this paper. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not reflect those of the sponsor.

6. REFERENCES

[1] Google search. http://www.google.com

[2] Yahoo! search. http://yahoo.com

[3] MSN search. http://www.msn.com

[4] Berger, A. and Lafferty, J. (1999). Information Retrieval as Statistical Translation. In *Proceedings of SIGIR '99*, 222-229.

[5] Inan, H., Search User Behavior Patterns. Available online at http://www.hurolinan.com/book/br_detail.asp?LocatorCode=229

[6] Teevan, J., Dumais, S. T. and Horvitz, E. (2005). Personalizing Search via Automated Analysis of Interests and Activities. In *Proceedings of SIGIR '05, 43-50.*

[7] Salton, G. and McGill, M. J. (1983). Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[8] Pitkow, J., Schutze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E. and Breuel, T. (2002). Personalized search. *Communications of the ACM, 45(9)*: 50-55.