Using Query Profiles for Clarification

Henning Rode and Djoerd Hiemstra

University of Twente, The Netherlands
{h.rode, d.hiemstra}@cs.utwente.nl

Abstract. The following paper proposes a new kind of relevance feedback. It shows how so-called query profiles can be employed for disambiguation and clarification.

Query profiles provide useful summarized previews on the retrieved answers to a given query. They outline ambiguity in the query and when combined with appropriate means of interactivity allow the user to easily adapt the final ranking. Statistical analysis of the profiles even enables the retrieval system to automatically suggest search restrictions or preferences. The paper shows a preliminary experimental study of the proposed feedback methods within the setting of TREC's interactive HARD track.

1 Introduction

When information retrieval left the library setting, where a user ideally could discuss her/his information need with a search specialist at the help-desk, many ideas came up how to imitate such interactive search scenario within retrieval systems. Belkin, among others, broadly sketches the system's tasks and requirements for interactive information seeking [1]. We do not want to further roll up the history of interactive information retrieval here, but to remind briefly its main aims.

In order to formulate clear queries, resulting in a set of useful, relevant answers, the user of a standard information retrieval system needs knowledge about the collection, its index, the query language and last but not least a good mental model of the searched object. Since it is unrealistic to expect such knowledge from an non-expert user, the system can assist the search process in a dialogue like manner. Two main branches of interactive methods try to bridge the gap between a vague information need and a precise query formulation:

Relevance Feedback helps the user refining the query without requiring sophisticated usage of the system's query language. Query terms are added or reweighted automatically by using the relevant examples selected by the user [2,3]. The examples shown to the user for judgement can either be documents, sentences out of those documents or even a loosely bundle of terms representing a cluster of documents. Experiments within TREC's interactive HARD track showed many variants of such techniques [4,5]. By presenting example answers to the user, relevance feedback can also refine the user's mental image of the searched object.

Browsing techniques, on the other hand, provide an overview on the existing document collection and its categorization (see e.g. the Open Directory Project [6]),

M. Lalmas et al. (Eds.): ECIR 2006, LNCS 3936, pp. 205–216, 2006.

or visualize the relation among documents [7]. The user can restrict the search to certain categories. This can also be regarded as a query refinement strategy. It is especially helpful, when the selected categorical restriction cannot be expressed easily by a few query terms.

The query clarification technique, we are proposing in this paper, belongs mainly to the first type, the relevance feedback methods. However, it combines the approach with summarization and overview techniques from the browsing domain. This way it tries not only to assist formulating the query, but also provides information about the collection in a query specific preview, the so-called *query profile*. Following an idea of Diaz and Jones [8] to predict the precision of queries by using their temporal profiles, we analyzed the application of different query profiles as an instrument of relevance feedback. The main aim of the profiles is to detect and visualize query ambiguity and to ask the user for clarification if necessary. We hope to enable the user to give better feedback by showing him/her this summarized information about the expected query outcome.

The paper is structured as follows: After a short look on two related approaches, we start in Sec. 2 by giving a definition of query profiles and explain how they can be generated. Sec. 3 discusses their application for query classification. Sec. 4 shows a possible score computation and combination to make use of the user feedback for an improved final ranking. We further present a preliminary experimental study of our relevance feedback technique and finish with conclusions about the achieved results.

1.1 Related Approaches

In order to distinguish our approach from similar ones, we finish this introduction by looking at two comparable methods. The first one is a search interface based on clustering suggested by Palmer et al. [9][1]. It summarizes results aiming at query disambiguation, but instead of using predefined categories as we will suggest for our topical profiles, it groups the documents using a not specified clustering algorithm. Whereas the clustering technique shows more topical adaptiveness, our static categories ensure always a useful grouping.

Another search interface proposed by Sieg et al. [10] assists the user directly in the query formulation process. The system compares the initial query with a static topic hierarchy and presents the best matching categories to the user for selecting preferences. The chosen categories are then used for query expansion. In contrast, our query profiles are not based on the few given query terms directly but on the results of an initial search. This way, we get a larger base for suggesting appropriate categories and we involve the collection in the query refinement process.

The mentioned approaches exclusively consider the topical dimension of the query. We will further discuss the usage and combination of query profiles on other document dimensions, in this case temporal query profiles.

---

[1] The one-page paper briefly explains the concept also known from the *Clusty* web search engine (`http://clusty.com`) coming from the same authors.

2 Query-Profiles

Looking from the systems perspective, the set of relevant answers to a given query is the set of the top ranked documents. This set can unfortunately differ by far from the set of documents relevant to the user. The basic idea of query profiles is to summarize information about the system's answer set in a suitable way to make such differences obvious.

Definition 1. *A* query profile *is the distribution of the top X ranked documents in the result set along a certain property dimension, like time, topic, location, or genre. E.g. a temporal query profile shows the result distribution along the time dimension, a topical profile along the dimension of predefined topics the documents belong to.*

The underlying assumption of the profile analysis is that clear queries result either in a profile with one distinctive peak or show little variance in case the property dimension is not important for the query. In contrast, we expect ambiguous queries to have query profiles with more than one distinctive peak.

Whereas the general ideas stay the same for all kinds of query profiles, there are several domain specific issues to consider. We will thus take a closer look on generating temporal and topical profiles, the two types used in the later experimental study.

2.1 Generating Temporal Profiles

Having a date-tagged corpus, a basic temporal profile for a given query is simple to compute. We treat the 100 top ranked documents D_j from the baseline run as the set of relevant answers and aggregate a histogram with monthly time steps H_i:

$$H_i = |\{D_j | month(D_j) = i\}| \ . \tag{1}$$

The decision for the granularity of one month is based on the overall time span of the corpus and the timeliness of news events. Other granularities, however, could be considered as well.

As a next step, we performed a *time normalization* on the profile. Knowing that the corpus articles are not evenly distributed over the total time span, the time profile should display the relative monthly frequency of articles relevant to the given topic rather than absolute numbers. Therefore, the frequency of each monthly partition H_i is divided by the total number of corpus articles C_i originating from month i. In order to avoid exceptional small numbers, the averaged monthly corpus frequency $avg(C)$ is used as a constant factor:

$$H_i^* = \frac{H_i}{C_i} * avg(C) \ . \tag{2}$$

Furthermore, we performed moving average smoothing on the histogram, a technique used for trend analysis on time series data [11]. It replaces the monthly frequencies of the profile by the average frequencies of a small time window around the particular month. We used here a window size of 3 months:

$$H_i^{**} = \frac{H_{i-1}^* + H_i^* + H_{i+1}^*}{3} \quad . \tag{3}$$

The graph in Fig. 1 shows an example of a resulting temporal profile. There are two reasons for using such a smoothing technique. First, the time-line the search topic is discussed in the news will often overlap with our casual monthly partitioning. Second, although we want to spot peaks in the profile, we are not interested in identifying a high number of splintered bursts. If two smaller peaks are lying in a near timely neighborhood they should be recognized as one.

Finally, we want to determine the number, bounds, and the importance of peaks in the temporal profile. Diaz and Jones [8] tried several techniques for this purpose and decided to employ the so-called burst model from Kleinberg [12]. It assumes a hidden state machine behind the random events of emitting the specific word in certain frequencies. The assumed machine changes over time between its norm and peak state, corresponding to phases with normal and high emission of the word respectively. The aim is then to find the unknown state sequence with the highest probability to cause the observed random events of the time profile. Kleinberg employs for this task the Viterbi algorithm.

We have used for the generation of temporal profiles a two state automaton $\mathcal{A}^2_{1.5}$ with a very low value for $\gamma \approx 0.02$[2]. The considerably different setting of parameters compared to Kleinberg's experiments can be explained by the fact that we analyzed profiles of word frequencies which are already averaged on the level of months. Hence bursts will remain smaller and less distinctive.

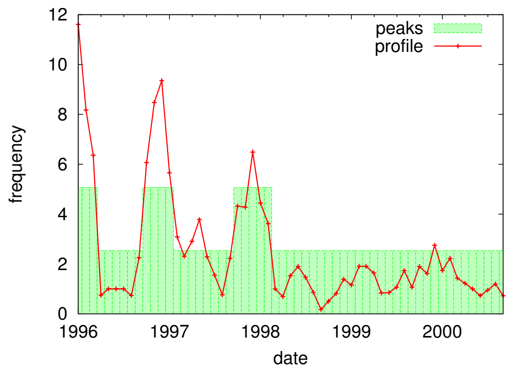


Fig. 1. Temporal Profile of Topic 363: *Transportation Tunnel Disasters*

When we also want to compute a measure for the importance of the found peaks P_j, the corresponding frequency values of the temporal profile can simply be summed up. A further division by the average of such frequency sums $avg(P)$ leads to a value for peak intensity better comparable among different temporal profiles:

$$P_j = \sum_{i \in range(P_j)} H_i^{**} \ , \qquad intensity(P_j) = \frac{P_j}{avg(P)} \quad . \tag{4}$$

2.2 Generating Topical Profiles

Generating topical profiles faces different issues than the ones explained for the temporal dimension. First and most important, the corpus is not topic-tagged.

[2] See [12] for a detailed description of the automaton and its parameters.

A topic classification is therefore required. Secondly, the topical dimension is not continuous but divided in a discrete set of previously defined concepts. In principle, topics could have a hierarchical relation but there won't be any natural definition of an order. So the identification of peak bounds as in the temporal dimension ceases to apply here.

For topic classification we need to build abstract models for all different concepts, the classification should take into account. Language models can be applied as classifiers for this purpose. In order to demonstrate the idea, we used models built on a different training corpus to distinguish 12 different topical concepts similar to the main sections of common newspapers, like politics or sports. A more detailed description about the construction of these language models can be found in [13].

The required text classification for computing a topical profile differs slightly from the typical categorization task (described in [14]). We do not need to assign binary labels whether a document belongs to a certain category or not. A similarity measure showing to which extend an article belongs to a given category is already sufficient. Hence, the task falls back to the known domain of ranking a set of documents given a query. In fact, an abstract language model describing a topical concept is nothing but an exceptional long query. We used in the experiments the NLLR measure (described in a later section) which is also applied to compute a score for the initial query. Only the smoothing factor λ is set smaller in this case. Firstly, because the exceptional query length makes smoothing less important, and secondly, to increase differences between the models.

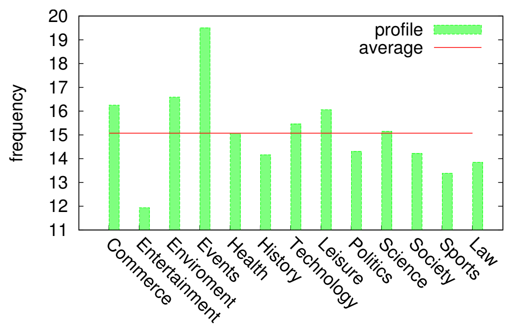


Fig. 2. Subject Profile of Topic 363: *Transportation Tunnel Disasters*

In order to speed up the computation of topical profiles as well as the later ranking procedure the score computation is performed off-line. For each classifier in the set of topical concepts a score vector is maintained, holding the individual scores for all documents within the collection. An example topical profile is displayed in Fig. 2.

After the classification task is done, topical profiles can be computed in the following way. Similar to temporal profiles explained previously, the set of the 100 top ranked documents given the query is determined. The score for a specific topic category T_i is then defined by the sum of all document scores from D for this category. The intensity value, as introduced in the last section, is computed accordingly:

$$T_i = \sum_{D_j} NLLR(T_i|D_j) \ , \qquad intensity(T_i) = \frac{T_i}{avg(T)} \ . \tag{5}$$

3 The Clarification Interface

After generating and analyzing the query profiles, we discuss in this section how the gained information can be presented to the user for query clarification. The user interface thereby has to fulfill two functions:

- It needs to present all necessary information to the user that allows her/him to take a decision.
- It should provide simple but powerful means to adapt the query in the intended way.

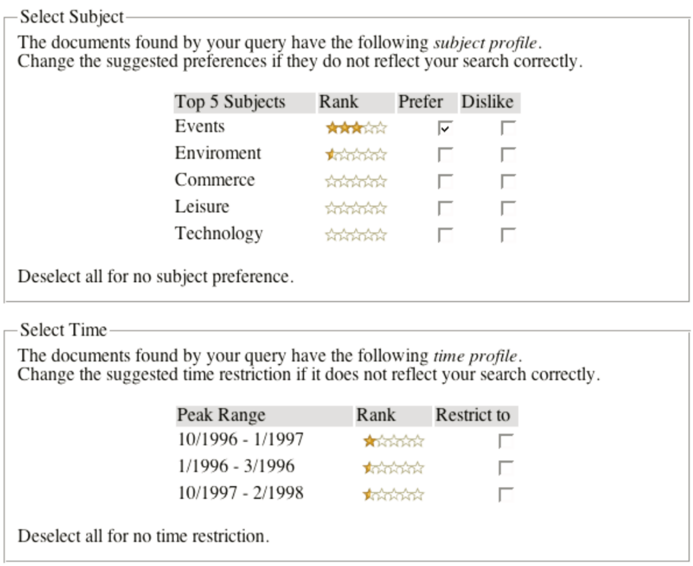
The second point needs further explanation. Not all search topics are easy to express by a few query terms. Although several articles contain the same keywords, their specific view on the topic or genre might not match the type of documents the user had in mind. If we allow the user to refine the query not only by further keywords but by selecting preferences to more abstract concepts or to restrict the search space to a certain location or time, the problem of expressing an information need accurately can be overcome. However, confronting a user in an advanced search interface with all possible combinations of restrictions and preferences to an in general unlimited number of concepts, dates, or locations, would overextend the searcher. Maybe he/she does not even know the correct query meta-data, e.g. the date or location of the event he/she is looking for. Query profiles can help here, since they allow to automatically find the most important meta-data concepts given the initial query terms. This way it is possible to provide the user with the necessary information to set preferences or restrictions and to limit the search dialog to the most interesting options.

Compared to the profiles shown in the last section (Fig. 1 and Fig. 2) a user does not need to see the whole spectrum of the profile. Instead it seems sufficient to cut out the most relevant part of it, which means the highest temporal or topical peaks. For the experiments, we just displayed the 5 top ranked topics, but all identified temporal peaks. In practice their number never exceeds 4. In order to demonstrate the usefulness of the profile information and to explain why we restrict the output to the top ranked parts of the profiles, let us distinguish three possible cases:

1. In case the initial query was clearly formulated, the user gets a positive confirmation by seeing the expected topic or time partition on top of the ranked profile list, succeeded by close related ones. The absence of non-matching topics will be enough information for the user here. He/she does not need to see a long list of minor ranking topics.
2. In case the query was ambiguous also unwanted topics or time partitions will populate the top of the ranked query profiles. In order to get an unambiguous output, it is now important to refine the query in a way that it excludes most of the unwanted answers, but keeps the relevant ones. Again, the end of the ranked profile list is less interesting, since the topics there are already efficiently excluded by the query.

3. In case the user does not even find the relevant topics or time partitions among the top part of the query profile, it won't help to just refine the query. Either the query needs to be reformulated entirely or the corpus does not include the documents the user is searching for.

The second case is the most interesting one since it requests appropriate query refinement strategies. Whereas a time restriction based on the profile can be expressed relatively easy, it is in general difficult for a user to find on his own additional keywords that allow to distinguish between the wanted and unwanted topics of the profiles. However, the system has already abstract classifiers at hand to perform such filtering. The simplest way to refine the query is thus to express preferences directly on the profile itself. For this reason we made our query profiles interactive by adding *prefer* and *dislike* buttons to the topic profiles and *restrict to* fields to the temporal profiles, refining the query in the obvious way. Their exact influence on the final ranking is discussed in the next section.



Select Subject

The documents found by your query have the following *subject profile*.
Change the suggested preferences if they do not reflect your search correctly.

Top 5 Subjects	Rank	Prefer	Dislike
Events		☑	☐
Enviroment		☐	☐
Commerce		☐	☐
Leisure		☐	☐
Technology		☐	☐

Deselect all for no subject preference.

Select Time

The documents found by your query have the following *time profile*.
Change the suggested time restriction if it does not reflect your search correctly.

Peak Range	Rank	Restrict to
10/1996 - 1/1997		☐
1/1996 - 3/1996		☐
10/1997 - 2/1998		☐

Deselect all for no time restriction.

Fig. 3. Experimental Clarification Form of Topic 363: *Transportation Tunnel Disasters*

3.1 Automatic Preselection

We also looked, whether it is possible to make an automatic suggestion of an appropriate selection in the profiles. Obviously, the most high ranked topics or temporal peaks are good candidates, especially if they distinctively stand off from the lower ranked ones. The intensity measure defined in the last section explicitly addresses these characteristics. Using an intensity threshold, we can preselect all topics and temporal peaks above[3]. These values have been shown

[3] In the experiments an intensity threshold of 1.2 was used for the topical profiles, respectively 1.5 for the temporal profiles.

high enough to assure the selection of only distinctive peaks of the profile. An example clarification form with preselected items is shown in Fig. 3.

Automatic preselection is especially helpful in the first of the three scenarios above where the query is unambiguous. In such a case user feedback is not necessary and the query refinement could be performed as a sort of "blind feedback" procedure to sharpen the topical or temporal focus.

4 Retrieval Model and Score Combination

In this section we show a possible score computation and combination taking into account the initial query as well as the preferences and restrictions stated in the query refinement process. The focus lies thereby on the issues of score normalization and combination. We have chosen a language modeling approach, however, in principle the proposed feedback technique could also be used in the setting of other retrieval models.

In particular, we employed the NLLR, the length-normalized logarithmic likelihood ratio [15], as a score function:

$$NLLR(Q|D) = \sum_{t \in Q} P(t|Q) * \log \left(\frac{(1-\lambda)P(t|D) + \lambda P(t|C)}{\lambda P(t|C)} \right) \ . \tag{6}$$

The additional factor λ below the fraction does not harm the ranking but ensures that documents having none of the query terms get a zero score.

The NLLR is able to compare query terms and documents as well as entire language models. Due to the normalization it produces comparable scores independent of the size of the query. Therefore it can be used for a document ranking given either a query or a topical language model. The factor λ determines the degree of smoothing with the background collection model. Since smoothing plays an important role for short queries, whereas it dilutes the score differences for large-scale query language models, this factor can be changed according to its application[4].

Next to the scoring itself, all single sources of relevance evidence need to be combined to one final ranking. We decided not to use query expansion techniques, but to combine the separately computed scores directly. This allows to make efficiently use of precomputed document scores for topic language models and avoids a second scoring of the initial query terms. When multiple preferences or dislikes have to be handled the logarithmic scores of their corresponding models M_i are simply added, respectively subtracted for disliked models:

$$m\text{-}score(D) = \sum_{M_i} NLLR(M_i|D) \ . \tag{7}$$

The final combination of the initial query score, called $q\text{-}score(D)$ now, and all summed up preference scores requires special attention. We have to ensure that

[4] We set λ to 0.85 for queries, but to 0.5 for topic models.

the scores on both sides deliver "compatible" values or even more to guarantee still the dominance of the initial query in the final result. A minimum-maximum normalization solves such a task (among others described in [16]). It shifts the minimum of a score range $min\_s = \min\{score(D^*)|D^* \in C\}$ to 0 and its maximum to 1. We further stressed the initial query by doubling its score value in the final ranking:

$$norm(score(D)) = \frac{score(D) - min\_s}{max\_s - min\_s} , \quad (8)$$

$$final\text{-}score(D) = 2 * norm(q\text{-}score(D)) + norm(m\text{-}score(D)) . \quad (9)$$

5 Experimental Study

We tried to evaluate our relevance feedback based on query profiles in the setting of the HARD track 2005. A set of 50 queries, which are regarded as difficult[5], is evaluated on a ≈ 2GB newspaper corpus, the Aquaint corpus. The track set-up allows one-step user interaction with so-called clarification forms that have to fit one screen and have to be filled out in less than 3 minutes. In the original TREC setting the sent-in clarification forms were filled out by the same person who later does the relevance assessments for the specific query. We repeated the experiment ourselves, asking different users to state preferences or restrictions in the clarification forms after reading the query description and query narrative coming with the TREC search topics. This way, we inevitably lose the consistency between clarification and relevance assessment ensured by the HARD setting. However, we could study differences in the user behavior and their results.

The 4 test users [6] have been shortly introduced to their task by demonstrating one randomly picked out example clarification form. They needed on average 35 min to accomplish the task of clarifying all 50 queries. We want to remark here, that the conducted experiment have to be regarded preliminary. It was not the intention to carry out a fully qualified user study, but to gather first indication whether the proposed feedback technique is able to improve retrieval.

In order to compare the improvements, we performed a *baseline* run using just the up to 3 words from the query title, further one run with the automatically derived preferences only as explained in Sec. 3, referred to as *automatic* run. From the 4 evaluated user runs, we present here the two most different to keep the figures clear. Whereas *user1* selected almost no topic dislikes, *user2* had the highest fraction of dislike statements among his topic preferences. For comparison, the *user2** run refers to the same user, but ignores his dislikes.

A closer look at the set of the 50 search topics revealed, that they have not been distinctive with respect to their temporal profile. In fact, there was

[5] The query set was taken from the Robust track which tries to tackle selected difficult queries in an ad hoc retrieval setting.

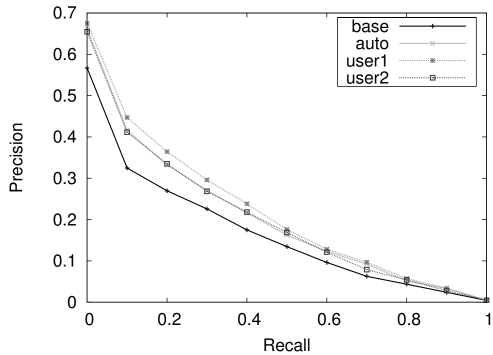
[6] 1 female – 3 male students, one of them working in computer science but not in the same project.

almost no case where the user wanted to restrict the query to a certain time span. Therefore, we restricted our analysis to the improvements by topical query refinement and ignored all temporal restrictions.

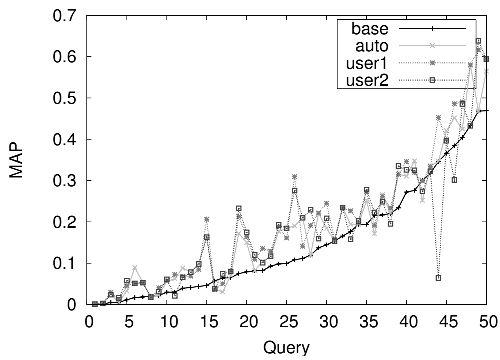
Fig. 4(a) presents an overview on the main evaluation measures computed for all presented runs. At a first glance it is obvious that the refined queries, even in our non-optimal evaluation setting, show a considerable improvement over the baseline run. The precision gain is most visible at the $P@10$ measures, which is an interesting characteristic aiming at a high precision at the top of the ranked list. The precision recall graph (Fig. 4(b)) confirms the observation made with the $P@10$ values. The precision gain stays the highest at the top of the ranked list. On the right side, the runs with query refinement slowly converge to the baseline, but always stay on top of it.

	base	*auto*	*user1*	*user2*	*user2**
MAP	0.151	0.187	0.204	0.187	0.201
R-Prec	0.214	0.252	0.268	0.255	0.265
P@10	0.286	0.380	0.396	0.354	0.402

(a) Result Overview



(b) Precision Recall Graph



(c) MAP Improvements on Single Queries

Fig. 4. Evaluation Results

The special run ignoring the topic dislikes of *user2* has a better general performance than its counterpart. Although it is not shown in the table, this observation holds for all four tested users. It indicates that topic dislike statements bear the risk to weaken the result precision in our current implementation.

Surprisingly, the values show also that the automatic run can compete with the user performed clarification. We cannot entirely explain this phenomenon, but can make two remarks on its interpretation. First, the query set has not been designed to test disambiguation. If a query asking for “Java” expects documents about the programming language, automatic topic feedback will work perfectly. However, it fails if in fact the island was meant. Examples of the second type are necessary to compare user and automatic feedback, but are unlikely to be included in the test set. A further reason for the good performance of the automatic run might simply be the fact that it did not contain dislike statements.

For a more detailed view on the results, Fig. 4(c) presents the evaluation of all single queries sorted by increasing MAP value of the baseline run. Thus, the graphic shows the worst performing queries on the left, continued by a section with still relatively low quality response in the middle, up to acceptable or even good queries on the right. Although the improvement per query is not stable, it seldom happens that the user feedback deteriorates the results. The one extreme case on the right side of the figure is again caused by dislike statements. If we consider the relative improvement, the queries in the middle part of the figure apparently gain the most from query refinement. Within the distinction of queries from Sec. 3 these queries probably fall under the ambiguous category 2. The fact that we encounter the highest improvement in this category nicely demonstrates the usefulness of our method.

6 Conclusions and Outlook

The results show promising improvements for all runs that make use of query profiles even in our preliminary experimental study. With a query set designed to test how retrieval systems cope with ambiguity, we would probably be able to show even higher improvements using our feedback method. The same applies for queries that reward temporal restrictions. Also a finer grained topical "resolution", potentially in form of a topic hierarchy, could lead to a more focused query profile on the topic dimension.

Further analysis is needed, how to involve topical dislike statements in a way that they do not harm the results, but also contribute to the query refinement. Furthermore, we need to examine query profiles on other dimensions. The temporal profiles remained untested by the current HARD track query set, but also geographical or genre profiles - in order to name just two possible other parameters - might enable similar improvements as the topical query refinement.

The automatic feedback method turned out to be an interesting side product of the work with query profiles. It performed almost as good as the user feedback. It raises the question to which extend the system can decide based on query profile statistics, whether automatic feedback is reliable enough in a certain case to omit user interaction. Especially when profiles on more dimensions get involved in the analysis, the user should not be bothered by a multiple number of feedback questions. Instead an intelligent retrieval system might be able to select the most helpful dimension for explicit user feedback itself.

References

1. Belkin, N.: Interaction with texts: Information retrieval as information-seeking behavior. In: Information Retrieval '93, Von der Modellierung zur Anwendung, Universitaetsverlag Konstanz (1993) 55–66
2. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. JASIS **41**(4) (1990) 288–297

3. Harman, D.: Relevance feedback revisited. In Belkin, N., Ingwersen, P., Pejtersen, A., eds.: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark, June 21-24, 1992, ACM (1992) 1–10
4. Allan, J.: Hard track overview in trec 2003: High accuracy retrieval from documents. In: TREC. (2003) 24–37
5. Allan, J.: Hard track overview in trec 2004: High accuracy retrieval from documents. In: TREC. (2004) 25–35
6. Netscape: Open directory project. `http://www.dmoz.org` (2005) Date retrieved: 03-11-2005.
7. Godin, R., Gecsei, J., Pichet, C.: Design of a browsing interface for information retrieval. In Belkin, N., Rijsbergen van, C., eds.: SIGIR'89, 12th International Conference on Research and Development in Information Retrieval, Cambridge, Massachusetts, USA, June 25-28, 1989, Proceedings, ACM (1989) 32–39
8. Diaz, F., Jones, R.: Using temporal profiles of queries for precision prediction. In Sanderson, M., Järvelin, K., Allan, J., Bruza, P., eds.: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Sheffield, UK (2004) 18–24
9. Palmer, C., Pesenti, J., Valdés-Pérez, R., Christel, M., Hauptmann, A., Ng, D., Wactlar, H.: Demonstration of hierarchical document clustering of digital library retrieval results. In: ACM/IEEE Joint Conference on Digital Libraries, JCDL 2001, Roanoke, Virginia, USA, June 24-28, 2001, Proceedings, ACM (2001) 451
10. Sieg, A., Mobasher, B., Lytinen, S., Burke, R.: Using concept hierarchies to enhance user queries in web-based information retrieval. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications. (2004)
11. Chatfiled, C.: The Analysis of Time Series. 3rd edition edn. Chapman and Hall (1984)
12. Kleinberg, J.: Bursty and hierarchical structure in streams. Data Mining and Knowledge Discovery **7**(4) (2003) 373–397
13. Rode, H., Hiemstra, D.: Conceptual Language Models for Context-Aware Text Retrieval. In: Proceedings of the 13th Text REtrieval Conference Proceedings (TREC). (2005)
14. Sebastiani, F.: Text categorization. In Zanasi, A., ed.: Text Mining and its Applications to Intelligence, CRM and Knowledge Management. WIT Press, Southampton, UK (2005) 109–129
15. Kraaij, W.: Variations on language modeling for information retrieval. PhD thesis, University of Twente, Netherlands (2004)
16. Croft, B.: Combining approaches to information retrieval. In Croft, B., ed.: Advances in Information Retrieval : Recent Research From the Center for Intelligent Information Retrieval. Kluwer Academic Publishers, New York (2002) 1–36