Query Phrase Suggestion from Topically Tagged Session Logs

Eric C. Jensen[1], Steven M. Beitzel[1], Abdur Chowdhury[2], and Ophir Frieder[1]

[1] Information Retrieval Laboratory, Illinois Institute of Technology,
10 W 31st St., Chicago, IL 60616, USA
{ej, steve, ophir}@ir.iit.edu
[2] America Online, Inc., 44100 Prentice Dr., Sterling, VA 20166, USA
Cabdur@aol.com

Abstract. Searchers' difficulty in formulating effective queries for their information needs is well known. Analysis of search session logs shows that users often pose short, vague queries and then struggle with revising them. Interactive query expansion (users selecting terms to add to their queries) dramatically improves effectiveness and satisfaction. Suggesting relevant candidate expansion terms based on the initial query enables users to satisfy their information needs faster. We find that suggesting query phrases other users have found it necessary to add for a given query (mined from session logs) dramatically improves the quality of suggestions over simply using cooccurrence. However, this exacerbates the sparseness problem faced when mining short queries that lack features. To mitigate this, we tag query phrases with higher level topical categories to mine more general rules, finding that this enables us to make suggestions for approximately 10% more queries while maintaining an acceptable false positive rate.

1 Introduction

Search system users typically pose short, vague queries and often revise them in an effort to find the desired, relevant results for their information need. Analysis of search session logs (containing the time-ordered queries submitted by a single user) has found that the average query is only 2.21 terms in length, but 33% of users reformulated their query at least once, with a mean of 2.84 queries per session [1]. Clearly, time and effort spent reformulating queries that yielded unsatisfactory results negatively impacts user satisfaction. This is exacerbated in environments such as a mobile interface where the interaction required to examine results and pose revised queries may be more tedious. Rather, a search interface capable of recognizing likely information needs for a vague query and suggesting relevant expansion terms to satisfy each of those needs is desirable.

It is clear that manual query refinement typically improves retrieval effectiveness. Kelly et al. shows a direct correlation between query length and effectiveness, finding that expanding queries using feedback from users via clarification forms significantly improved their performance over 45 TREC-style topics [2]. It is also well-documented that users are not often sure of the correct terms for

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 185–196, 2006.

which to describe their information need [3]. As such, it is of interest to search services to be able to suggest additional terms and phrases to their users. For example, a vague query such as "tickets" might be quickly narrowed to satisfy a specific need by suggesting the terms "baseball," "concert," and "traffic." These suggestions serve as both clarifying aids to users, and a hint as to how the system interprets their queries as initially expressed. Studies have shown that users are more satisfied when they at least understand why a search system produced the given results [4]. Expansion suggestions help users to satisfy their information needs faster by reducing the interaction required to express them.

As with the search problem itself, term suggestion is complicated by the fact that typical, short queries on their own do not carry many features from which to determine relevant suggestions. The Zipfian nature of word frequencies is well known. The popularity of queries and query terms follows a similar distribution [5]. These studies have also shown that the population of queries is generally tail-heavy, with almost 50% of all queries occurring five times or less in a week. This preponderance of rare queries makes it difficult to base suggestions on the frequency of their term appearances and pure cooccurrence with query terms alone. If so many queries occur this rarely, the number of times users add the same expansion terms to those queries is an order of magnitude more rare. Nevertheless, this large number of previously unseen or very rarely seen queries must be addressed when doing suggestion. The intuition that these may be more descriptive and not need suggestions is not necessarily true, as there are many proper nouns, etc. that are rarely seen but nonetheless vague or ambiguous. For example, if a user searches for "fqas" it is likely to be a rare query. Any system that relies solely on exactly what other terms cooccur with "fqas" is not likely to be very helpful. This suggests that we must take a deeper look at the context in which phrases occur: the revisions other users have gone through in their sessions with similar queries.

The fundamental obstacle to mining session logs for suggestions is the increasing sparseness of the feature space as one moves from terms to queries to query revisions. As such, this paper has two main contributions. First, we mine suggestions from only phrases other users actually added for a given query. This dramatically improves the quality of suggestions over simply finding "related" cooccurring query terms. For example, the terms "baseball" and "game" frequently cooccur, but "game" is not a very useful term for narrowing the query "baseball" to a specific information need. "tickets", "scores," or "players" are more likely to get the user to the information they desire faster. Focusing on mining suggestions only from terms other users have actually used to "narrow" their query moves away from simple relatedness and focuses on actual likely user behavior. For example, a user might have a narrow session such as "new york baseball → yankees", meaning that they first searched for "new york baseball", and decided to narrow their scope by adding "yankees" later on in their session. We analyze these "narrow" sessions and use them to suggest terms for interactive query expansion. While this provides better suggestions, it does not address the previously mentioned issue of making suggestions for queries that are rare, but

still in need of relevant suggestions. To mitigate this, we also map query terms to more abstract classes in order to mine patterns with any reasonable support frequency. We propose applying topical tags such as "MUSICIAN," "CITY," or "VIDEO_GAMES" to query phrases to achieve more general rules such as "US_LOCALITIES high → school". Because queries are short, however, traditional entity tagging is difficult to apply. Rather, we use manually-edited lists to identify topical tags. We evaluated these techniques using 44,918 sessions from a large AOL™search query log and manual evaluated over 7000 suggestions for 353 queries (the largest such manual evaluation we are aware of, available at http://ir.iit.edu/collections).

2 Prior Work

Leveraging users' reformulations depends on the ability to accurately partition a query log into user sessions. Murray, et al. examine how the method of session detection influences applications using those sessions, pointing out that defining sessions by a single timeout removes the possibility of studying meaningful time gaps between particular users' queries [7].

While term suggestion based on fixed vocabularies, or thesauri, are feasible in focused, relatively static environments [8], they are not scalable to large, dynamic environments such as the web. Several studies propose term suggestion based on query clustering. Wen et al. used direct clickthrough data as well as grouping of queries that return similar documents to perform a kind of query clustering, and showed that using this combined method outperforms both methods individually [9]. Baeza-Yates et al. cluster queries from a session log based on the terms in clicked documents and achieves 80% precision in their top three suggested queries for ten candidate queries from a Chilean query log [10] although again, no results using this clustering technique to aid in reformulation suggestion are provided.

Fonseca et al. found that when users manually expanded 153 queries with concepts mined from associated queries in a session log a 32-52% relative improvement in retrieval average precision was obtained [11]. Kawamae et al. defines a user behavior model quantifying the number of specializations, generalizations, etc. in a query log [12]. Jones and Fain quantify similar user behaviors from a log and find they can predict which query terms will be deleted in a session with 55.5% accuracy over 2000 queries [13]. These studies work to quantify the ways that users behave when they engage in query reformulation, but to our knowledge there have been no studies that make direct use of the actual terms added by users when attempting to narrow the scope of their queries. Huang et al. found that single terms suggested based on a cooccurrence matrix mined from Chinese query logs (in which phrases are less important than English ones) had much higher precision than those suggested based on retrieved documents over 100 queries [14]. Other studies focus on the more specific goal of suggesting terms to advertisers in a pay-per-performance search market. Gleich et al. used SVD to suggest terms from an advertising keyword database and achieved over 80% precision at up to 40% recall levels on the top 10,000 suggested terms for two queries [15].

Herlocker et al. reviews many metrics used to evaluate suggestion systems, dividing them into equivalence classes [16]. They also make the important distinction that this task must consider coverage over a representative sample of queries in addition to traditional precision. A proper evaluation requires "offline" evaluations on existing datasets in addition to live-user or "online" evaluations because it is often not feasible to manually evaluate enough queries for a successful online evaluation. If the sample of judged queries is too small, an online evaluation using the sample will not give a reliable measure of the true proportion of queries for which suggestions are being offered.

3 Methodology

Providing relevant suggestions for interactive query expansion depends on the ability to distinguish terms that focus the query on likely information needs from those that are simply related, and the ability to abstract rare queries and query terms to support suggestions for them. To address each of these issues, we propose two techniques. 1. Mining suggestions from other users' behavior in a query session log. 2. Tagging query phrases with topical categories to mine more general rules. As a baseline for the first technique, we use the simple cooccurrence of phrases in queries to make suggestions. Because they are non-exclusive, the gain in effectiveness when combining the second technique with the first is then measured rather than applying it to the poorer baseline.

All techniques use the same function for scoring candidate suggestions. There are many metrics for measuring the interestingness of cooccurrences or association rules. We initially chose frequency-weighted mutual information (FWMI) because of its simplicity and emphasis on the frequency of cooccurrences in addition to normalization by the probability of either term occurring independently [17]. However, we found, as they did, that a small number of very frequently occurring terms were scored too highly, producing irrelevant associations. We therefore weighted with the logarithm of the frequency instead (Equation 1) which performs much better for all techniques and is commonly done in information retrieval.

$$LFWMI(q,s) = \log_2(C(q \rightarrow s)) \log_2 \frac{P(q \rightarrow s)}{P(q)P(s)} \tag{1}$$

Equation 1 scores each candidate phrase for suggestion, s, with respect to a single query phrase, q. This would be sufficient if we treat each query as a single q, but to mine sufficient evidence to make suggestions for the large numbers of rare queries (as discussed in section 1), we must decompose each query Q into its component phrases q_i. In our experimentation, we limit phrases to term unigrams and bigrams, ignoring the position in which terms occur to further aggregate evidence. However, the cooccurrence counts from our training data enable us to intelligently chunk the query, so that when processing the query "new york state" we use the same mutual information metric to identify it as consisting of two phrases "new york" and "state" and not treat the unigrams "new" and "york"

independently. After chunking the query, we average the mutual information for each candidate suggestion across the query's component phrases (Equation 2) to obtain the final score.

$$Score(Q, s) = \frac{\sum_{q_i \in Q} LFWMI(q_i, s)}{|Q|} \quad (2)$$

When mining the session log, we also decompose the sets of terms users add for a given query, so that we can combine evidence from sessions such as "new york → state college" and "new york → university state" to find that the unigram "state" is often added to queries containing the bigram "new york". When making suggestions, we do not suggest unigrams that also exist in bigram suggestions unless they are scored higher. Finally, we select a threshold minimum score required to make a suggestion, filtering out any suggestions below that threshold.

This same processing is done for all of our techniques. Only the definition of q_i and s varies. For the baseline cooccurrence technique, all unigram and bigram q_i are counted as candidate suggestions s for one another. At suggestion time, the same phrase based chunking and averaging above is applied. When using only those terms other users have actually applied to "narrow" their query, only the actual "narrow" unigrams and bigrams that are added in the subsequent query are counted as candidate suggestions s for the initial query phrases q_i. When topical tagging is applied, phrases are replaced with their tags and narrow terms are counted as candidate suggestions for unigrams and bigrams of the tagged query. For example, the session "new york state → college" would count "college" as a candidate suggestion for "US_STATE state" (a state name followed by the term "state") and for "US_STATE" alone. We combine suggestions from topical tagging and the terms alone by simply summing their scores if the same phrase is suggested by both, weighting the topically tagged scores at 20% of the literal ones because their generality makes them noisier.

4 Results

The definition of relevance in a phrase suggestion task is different from that in the information retrieval problem. Rather than assuming that the query represents a single information need and seeking to find documents relevant to that need, the suggestion task assumes a query could represent multiple information needs. A relevant suggestion, then, is one which would help to narrow the query into a particularly likely information need. We begin with a description of our session log and manual evaluation. Section 4.2 compares using only those phrases users actually add to narrow their queries versus a baseline of simply using cooccurring phrases. Section 4.3 goes on to combine suggestions based on topically tagging queries with those from the terms themselves.

4.1 Experimental Environment

We performed our experiments using session data from AOL™search query logs. A random sample of session data containing approximately five million

queries from November 2004 through January 2005 was collected. The logs contain anonymous user ID's and the query terms that were used in the search. In total, there were 268,023 unique user ID's in the log, and approximately 28.4% of all queries were part of some kind of reformulation or revision (a similar finding to [1]). As discussed in section 2 there are many ways to define and determine session breaks. For the purposes of training and testing our techniques, we use only sequences of queries for which the same user inputs an initial query and then adds terms to that query without removing or changing any of the initial terms ("narrowing" their query). When the user makes multiple revisions of that type, we use only the terms in their initial and final query, with the assumption that if they succeeded in fulfilling their information need, it was with their final try. Obviously, suggestions should be useful for any initial query, but training and testing using cases where users delete some of their initial terms would complicate our evaluation. We trained our system using a random sample of 2/3 of the users in the log. The remaining 1/3 of the log was used for testing. It was comprised of 89,203 user ID's with 44,918 instances of pure "narrow" behavior.

Two evaluations are presented in parallel. The first is an "online" evaluation in which we had student assessors manually judge each suggestion as either relevant or not relevant for 353 distinct randomly sampled queries in the test 1/3 of the log. We pooled all suggestions from each of our techniques (21.5 per query on average) so that every suggestion evaluated was explicitly judged relevant or not relevant. On average, 17.8% of each pool (3.83 suggestions) were judged relevant. For 140 queries we had multiple assessors judge the same suggestions. Over these queries, 37% of the individual suggestion judgments disagreed. We had assessors rejudge that portion to form a single set of judgments for the rest of our evaluation. The second evaluation is an "offline" evaluation in which we treat only those terms the user actually added to a session (the "narrow" terms) as relevant and measure effectiveness over the entire test 1/3 of the log. Because this portion of the log contains all occurrences of queries, popular queries figure more heavily into the average performance than in the online evaluation which treats each query with equal weight. Also, the offline evaluation assumes only the actual phrase the user added is relevant. Therefore its associated error probabilities are very much inflated compared to the online one.

In both evaluations, we only examine the top five suggested phrases (ranked by score), with the intuition that suggesting a large number of phrases is not likely to be useful if the user must scroll through them to find an appropriate one. Different suggestion applications likely have different costs for retrieving irrelevant suggestions and missing relevant suggestions. Therefore, we apply filtering-style evaluation metrics, examining the probability of each of these two errors for varying thresholds on DET curves as is typically done in the Topic Detection and Tracking (TDT) conference [18]. To examine individual points (such as the optimal recall points in the tables below) we similarly combine the costs of these errors as is done at TDT (Equation 3) [19]. We estimate the probability of a suggestion being relevant for each query using the ratio of the number of relevant suggestions for that query to the maximum number of evaluated suggestions

across all queries (65). This probably underestimates P(rel), disproportionately weighting P(miss). In our online evaluation, we set C_{miss} to be 10 times, and in the offline evaluation 20 times, that of C_{fa}, reflecting our focus on recall and correcting for the bias in P(rel).

$$Cost = C_{miss}P(miss)P(rel) + C_{fa}P(fa)(1 - P(rel)). \quad (3)$$

4.2 Session Narrows vs. Simple Cooccurrence

First, we examine the effectiveness of using phrases users actually added to their queries versus simply how many times phrases cooccurred in general. As can be seen by the online and offline evaluations in Figure 1 and Figure 2, using actual terms users added is much more effective. As is traditional with DET curves, these graphs are on log-log scales [18]. In the online evaluation, we see that at lower thresholds simple cooccurrence does not find more relevant suggestions, it just retrieves many bad ones. In the offline one, the miss rate continues to decrease as a larger number of rare queries are given suggestions. Some of the top associations found from other users "narrow" behavior are "blue book → kelly," "paul johnson → beheading" (because it was in the news at the time our log was collected), and "costumes → halloween."

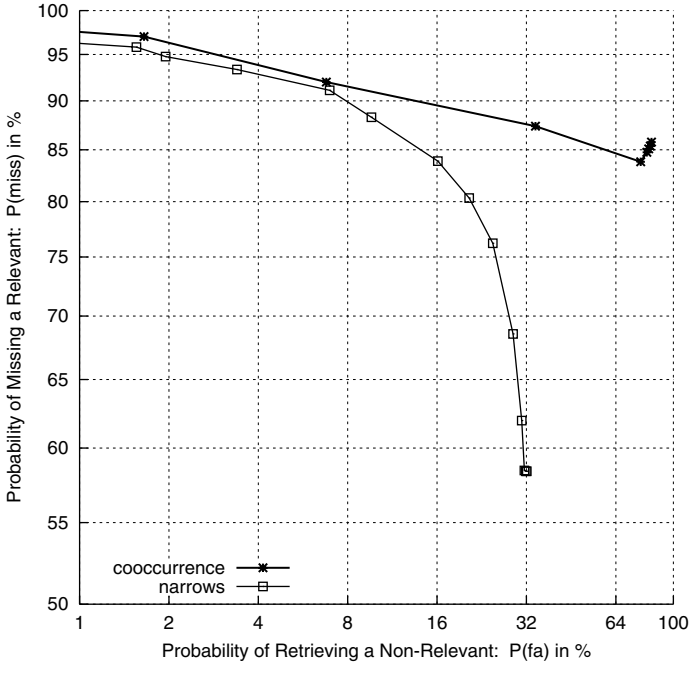

Fig. 1. Online Evaluation of Session Narrows vs. Simple Cooccurrence

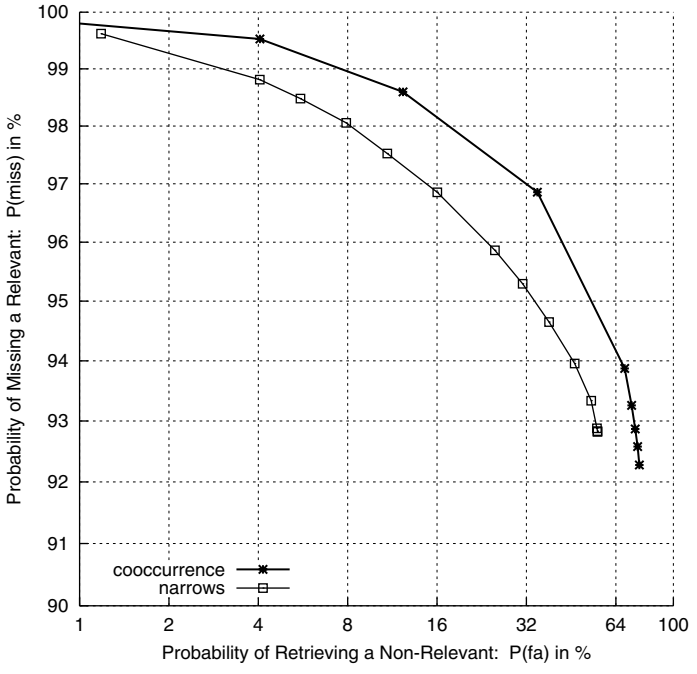

Fig. 2. Offline Evaluation of Session Narrows vs. Simple Cooccurrence

Table 1. Optimal Recall Points for Session Narrows vs. Simple Cooccurrence

Metric	Cooccurrence	Narrows	Improvement
Online P(fa)	77.62	32.08	45.540
Online P(miss)	83.81	58.39	25.420
Online Avg. Cost	2.096	1.191	0.905
Queries w/ Suggestion	353	312	-41 (-11.6%)
Offline P(fa)	76.80	55.54	21.260
Offline P(miss)	92.28	92.82	-0.540
Offline Avg. Cost	19.223	19.120	0.103
Queries w/ Suggestion	33425	27764	-5661 (-16.9%)

As we saw in Figure 1 and Figure 2, mining the phrases other users have actually used to narrow their queries dramatically outperforms cooccurrence. However, both techniques are still missing over 50% of the relevant results. All but the worst cooccurrence false alarm rates are likely acceptable as they are not far from the 37% assessor disagreement we found when judging. Table 1 contains the optimal recall points for each technique. While cooccurrence enables suggestions for more queries, it comes at a cost of 21% absolute increase in offline false alarms to gain 0.5% reduction in misses. Note that the number of queries

with a suggestion for the offline evaluation is out of the 44,918 total in the 1/3 test portion of our log.

4.3 Session Narrows vs. Category Tagged Narrows

Next, we examine the performance of using the "narrow" terms users added for particular query terms versus that of also combining suggestions based on tagging the queries with topical categories. Topical tagging of query phrases is achieved using manually edited lists of phrases for 256 categories ranging in specificity and totalling 1.25 million phrases created by AOL^{TM}editors. Some of the top ranked narrows for topically tagged queries are "VIDEO_GAMES → xbox," "COUNTIES election → results" and "GARDEN high → school" (because GARDEN includes plant names such as redwood and laurel). The online and offline evaluations in Figure 3 and Figure 4 show that we can achieve substantially higher recall using the topical category tags. These graphs focus only on the high-recall (low probability of missing a relevant result) portion of the curves. The higher precision performance of topically tagging is equivalent to that of the terms alone. This is because we weight the suggestions from topical tagging at only 20% of those from the terms themselves when combining them, essentially placing the suggestions from topical tagging below any found

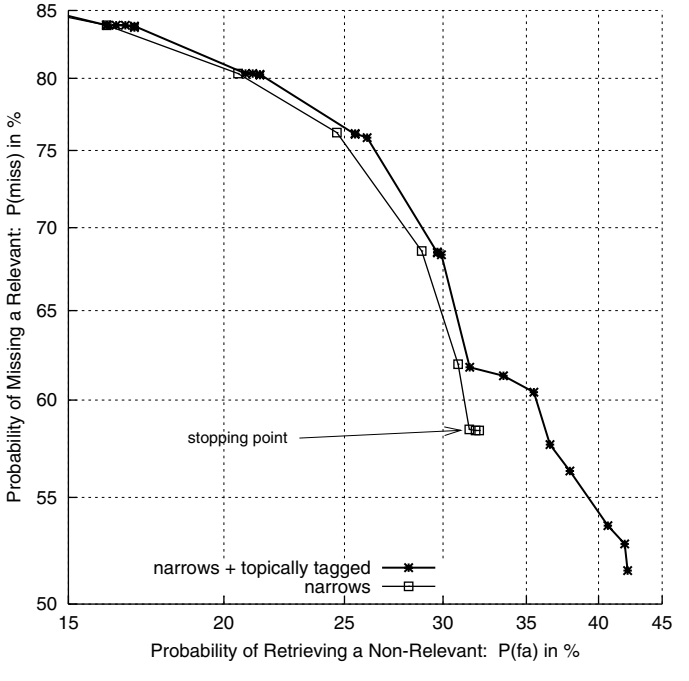

Fig. 3. Online Evaluation of Session Narrows vs. Narrows + Topically Tagged Narrows

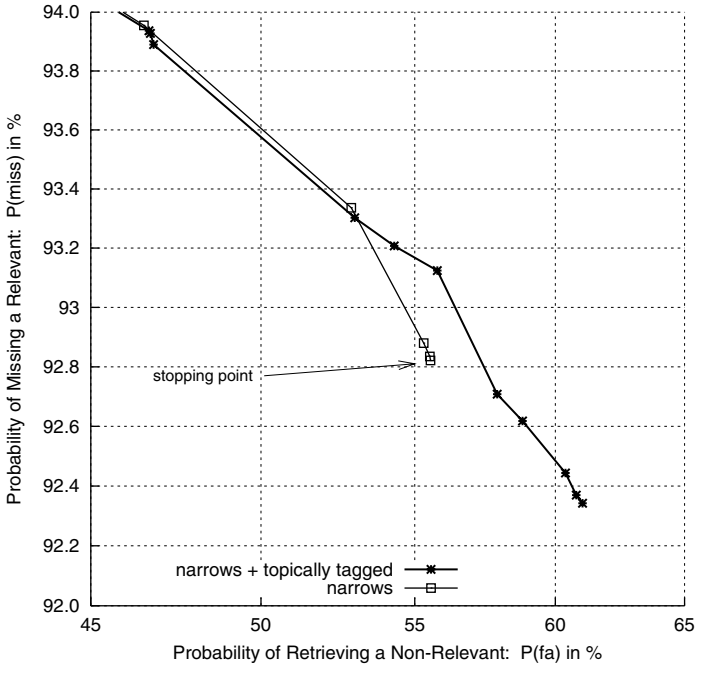

Fig. 4. Offline Evaluation of Session Narrows vs. Narrows + Topically Tagged Narrows

from the terms in the majority of cases. Higher weights for the suggestions from topical tagging produced inferior results, even when restricting suggestions from topical tagging to higher score thresholds than those for the terms themselves. This is likely because suggestions based on the specific terms are generally more reliable than those based on generalizations of the terms; when a suggestion is found using the specific terms themselves, suggestions from topical tagging can only add noise. Finding the same suggestion from topical tagging and the

Table 2. Optimal Recall Points for Narrows vs. Narrows + Topically Tagged Narrows

Metric	Narrows	Tagged	Improvement
Online P(fa)	32.08	42.21	-10.130
Online P(miss)	58.39	51.52	6.870
Online Avg. Cost	1.191	1.174	0.017
Queries w/ Suggestion	312	344	32 (10.3%)
Offline P(fa)	55.54	61.02	-5.480
Offline P(miss)	92.82	92.34	0.480
Offline Avg. Cost	19.120	19.079	0.041
Queries w/ Suggestion	27764	30402	2638 (9.5%)

terms themselves cannot be treated as multiple evidence of that suggestion being relevant.

Again, the benefit of using topical tagging is the ability to make relevant suggestions for a larger number of queries (to lower the probability of a miss). Whereas using only the phrases themselves cannot go below a "stopping point" of miss rate, no matter how low we set the threshold, topical tagging reaches nearly 7% below this in online miss rate, and reduces offline misses by nearly 0.5% with a cost of only approximately 5% increase in false alarms (as compared to 21% increase required for similar miss reduction when using cooccurrence). This stopping point for the phrases themselves is a direct result of the sparsity problem. No matter what quality of results are acceptable, there simply isn't evidence for making any suggestion for phrases that haven't been previously seen. Table 2 contains the optimal recall points for each technique. Using topical tagging we can increase recall while maintaining an acceptable false alarm rate.

5 Conclusion

It has been shown that searchers struggle in formulating the most effective queries for their information needs. Although it has also been documented that interactive query expansion by the user dramatically improves effectiveness, prior studies on making query reformulation suggestions using simple term cooccurrence have had only limited success. We have developed a technique for making suggestions that works by mining the terms other users actually added to a given initial query, improving the relevance of suggestions dramatically over simply using cooccurrence. We have also found that abstracting these queries with topical tags is effective in achieving substantially higher recall, allowing us to make suggestions for approximately 10% more queries while maintaining an acceptable false-positive rate. An obvious avenue for future work is to find a more principled way to combine evidence from each query phrase for a given association.

References

1. Spink, A., Jansen, B.J., Ozmutlu, H.C.: Use of query reformulation and relevance feedback by excite users. Internet Research: Electronic Networking Applications and Policy **10**(4) (2000) 317–328
2. Kelly, D., Dollu, V.D., Fu, X.: The loquacious user: A document-independent source of terms for query expansion. In: ACM Conference on Research and Development in Information Retrieval. (2005)
3. Belkin, N.J.: The human element: Helping people find what they don't know. Communications of the ACM **43**(8) (2000) 58–61
4. Hersh, W.: Trec 2002 interactive track report. In Voorhees, E.M., Buckland, L.P., eds.: Proceedings of the Eleventh Text Retrieval Conference (TREC 2002). Volume SP 500-251., NIST (2002)
5. Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O.: Hourly analysis of a very large topically categorized web query log. In: ACM SIGIR Conference on Research and Development in Information Retrieval. (2004) 321–328

6. Shen, X., Tan, B., Zhai, C.: Context sensitive information retrieval using implicit feedback. In: ACM Conference on Research and Development in Information Retrieval. (2005)
7. Murray, G.C., Lin, J., Chowdhury, A.: Characterizing web search user sessions with hierarchical agglomerative clustering. In: forthcoming. (2006)
8. Sihvonen, A., Vakkari, P.: Subject knowledge, thesaurus-assisted query expansion and search success. In: RIAO. (2004)
9. Wen, J.R., Zhang, H.J. Information Retrieval and Clustering. In: Query Clustering in the Web Context. Kluwer Academic Publishers (2003) 195–226
10. Baeza-Yates, R., Hurtado, C., Mendoza, M.: Query recommendation using query logs in search engines. In: International Workshop on Clustering Information over the Web. (2004)
11. Fonseca, B.M., Golgher, P., Pssas, B., Ribeiro-Neto, B., Ziviani, N.: Concept based interactive query expansion. In: ACM Conference on Information and Knowledge Management. (2005)
12. Kawamae, N., Takeya, M., Hanaki, M.: Semantic log analysis based on a user query behavior model. In: IEEE International Conference on Data Mining. (2003)
13. Jones, R., Fain, D.C.: Query word deletion prediction. In: ACM Conference on Research and Development in Information Retrieval. (2003)
14. Huang, C.K., Chien, L.F., Oyang, Y.J.: Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society of Information Science and Technology **54**(7) (2003) 638649
15. Gleich, D., Zhukov, L.: Svd based term suggestion and ranking system. In: IEEE International Conference on Data Mining. (2004)
16. Herlocker, J.L., Kostan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems **22**(1) (2004) 553
17. Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
18. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The det curve in assessment of detection task performance. In: Proceedings of the 5th ESCA Conference on Speech Communication and Technology (Eurospeech '97). (1997) 1895–1898
19. Manmatha, R., Feng, A., Allan, J.: A critical examination of tdt's cost function. In: Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval. (2002) 403–404