Robust Classification of Rare Queries Using Web Knowledge

Andrei Broder, Marcus Fontoura, Evgeniy Gabrilovich,
Amruta Joshi, Vanja Josifovski, Tong Zhang
Yahoo! Research, 2821 Mission College Blvd, Santa Clara, CA 95054
{broder | marcusf | gabr | amrutaj | vanjaj | tzhang}@yahoo-inc.com

ABSTRACT

We propose a methodology for building a practical robust query classification system that can identify thousands of query classes with reasonable accuracy, while dealing in real-time with the query volume of a commercial web search engine. We use a blind feedback technique: given a query, we determine its topic by classifying the web search results retrieved by the query. Motivated by the needs of search advertising, we primarily focus on rare queries, which are the hardest from the point of view of machine learning, yet in aggregation account for a considerable fraction of search engine traffic. Empirical evaluation confirms that our methodology yields a considerably higher classification accuracy than previously reported. We believe that the proposed methodology will lead to better matching of online ads to rare queries and overall to a better user experience.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval— *relevance feedback, search process*

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Query classification, Web search, blind relevance feedback

1. INTRODUCTION

In its 12 year lifetime, web search had grown tremendously: it has simultaneously become a factor in the daily life of maybe a billion people and at the same time an eight billion dollar industry fueled by web advertising. One thing, however, has remained constant: people use very short queries. Various studies estimate the average length of a search query between 2.4 and 2.7 words, which by all accounts can carry only a small amount of information. Commercial search engines do a remarkably good job in interpreting these short strings, but they are not (yet!) omniscient. Therefore, using additional external knowledge to augment the queries can go a long way in improving the search results and the user experience.

At the same time, better understanding of query meaning has the potential of boosting the economic underpinning of Web search, namely, online advertising, via the *sponsored search* mechanism that places relevant advertisements alongside search results. For instance, knowing that the query "SD450" is about cameras while "nc4200" is about laptops can obviously lead to more focused advertisements even if no advertiser has specifically bidden on these particular queries.

In this study we present a methodology for *query classification*, where our aim is to classify queries onto a commercial taxonomy of web queries with approximately 6000 nodes. Given such classifications, one can directly use them to provide better search results as well as more focused ads. The problem of query classification is extremely difficult owing to the brevity of queries. Observe, however, that in many cases a human looking at a search query and *the search query results* does remarkably well in making sense of it. Of course, the sheer volume of search queries does not lend itself to human supervision, and therefore we need alternate sources of knowledge about the world. For instance, in the example above, "SD450" brings pages about Canon cameras, while "nc4200" brings pages about Compaq laptops, hence to a human the intent is quite clear.

Search engines index colossal amounts of information, and as such can be viewed as very comprehensive repositories of knowledge. Following the heuristic described above, we propose to use the search results themselves to gain additional insights for query interpretation. To this end, we employ the pseudo relevance feedback paradigm, and assume the top search results to be relevant to the query. Certainly, not all results are equally relevant, and thus we use elaborate *voting schemes* in order to obtain reliable knowledge about the query. For the purpose of this study we first dispatch the given query to a general web search engine, and collect a number of the highest-scoring URLs. We crawl the Web pages pointed by these URLs, and classify these pages. Finally, we use these result-page classifications to classify the original query. Our empirical evaluation confirms that using Web search results in this manner yields substantial improvements in the accuracy of query classification.

Note that in a practical implementation of our methodology within a commercial search engine, all indexed pages can be pre-classified using the normal text-processing and indexing pipeline. Thus, at run-time we only need to run the voting procedure, without doing any crawling or classification. This additional overhead is minimal, and therefore

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

the use of search results to improve query classification is entirely feasible in run-time.

Another important aspect of our work lies in the choice of queries. The volume of queries in today's search engines follows the familiar power law, where a few queries appear very often while most queries appear only a few times. While individual queries in this long tail are rare, together they account for a considerable mass of all searches. Furthermore, the aggregate volume of such queries provides a substantial opportunity for income through on-line advertising.[1]

Searching and advertising platforms can be trained to yield good results for frequent queries, including auxiliary data such as maps, shortcuts to related structured information, successful ads, and so on. However, the "tail" queries simply do not have enough occurrences to allow statistical learning on a per-query basis. Therefore, we need to aggregate such queries in some way, and to reason at the level of aggregated query clusters. A natural choice for such aggregation is to classify the queries into a topical taxonomy. Knowing which taxonomy nodes are most relevant to the given query will aid us to provide the same type of support for rare queries as for frequent queries. Consequently, in this work we focus on the classification of rare queries, whose correct classification is likely to be particularly beneficial.

Early studies in query interpretation focused on query augmentation through external dictionaries [22]. More recent studies [18, 21] also attempted to gather some additional knowledge from the Web. However, these studies had a number of shortcomings, which we overcome in this paper. Specifically, earlier works in the field used very small query classification taxonomies of only a few dozens of nodes, which do not allow ample specificity for online advertising [11]. They also used a separate ancillary taxonomy for Web documents, so that an extra level of indirection had to be employed to establish the correspondence between the ancillary and the main taxonomies [18].

The main contributions of this paper are as follows. First, we build the query classifier *directly* for the target taxonomy, instead of using a secondary auxiliary structure; this greatly simplifies taxonomy maintenance and development. The taxonomy used in this work is two orders of magnitude larger than that used in prior studies. The empirical evaluation demonstrates that our methodology for using external knowledge achieves greater improvements than those previously reported. Since our taxonomy is considerably larger, the classification problem we face is much more difficult, making the improvements we achieve particularly notable. We also report the results of a thorough empirical study of different voting schemes and different depths of knowledge (e.g., using search summaries vs. entire crawled pages). We found that crawling the search results yields deeper knowledge and leads to greater improvements than mere summaries. This result is in contrast with prior findings in query classification [20], but is supported by research in mainstream text classification [5].

2. METHODOLOGY

Our methodology has two main phases. In the first phase, we construct a *document classifier* for classifying search results into the same taxonomy into which queries are to be classified. In the second phase, we develop a *query classifier* that invokes the document classifier on search results, and uses the latter to perform query classification.

2.1 Building the document classifier

In this work we used a commercial classification taxonomy of approximately 6000 nodes used in a major US search engine (see Section 3.1). Human editors populated the taxonomy nodes with labeled examples that we used as training instances to learn a document classifier in phase 1.

Given a taxonomy of this size, the computational efficiency of classification is a major issue. Few machine learning algorithms can efficiently handle so many different classes, each having hundreds of training examples. Suitable candidates include the nearest neighbor and the Naive Bayes classifier [3], as well as prototype formation methods such as Rocchio [15] or centroid-based [7] classifiers. A recent study [5] showed centroid-based classifiers to be both effective and efficient for large-scale taxonomies and consequently, we used a centroid classifier in this work.

2.2 Query classification by search

Having developed a document classifier for the query taxonomy, we now turn to the problem of obtaining a classification for a given query based on the initial search results it yields. Let's assume that there is a set of documents $D = d_1 \dots d_m$ indexed by a search engine. The search engine can then be represented by a function $\vec{f} = similarity(q, d)$ that quantifies the affinity between a query q and a document d. Examples of such affinity scores used in this paper are **rank**—the rank of the document in the ordered list of search results; **static score**—the score of the goodness of the page regardless of the query (e.g., PageRank); and **dynamic score**—the closeness of the query and the document.

Query classification is determined by first evaluating conditional probabilities of all possible classes $P(C_j|q)$, and then selecting the alternative with the highest probability $C_{max} = arg\max_{C_j \in C} P(C_j|q)$. Our goal is to estimate the conditional probability of each possible class using the search results initially returned by the query. We use the following formula that incorporates classifications of individual search results: $P(C_j|q) =$

$$\sum_{d \in D} P(C_j|q,d) \cdot P(d|q) = \sum_{d \in D} \frac{P(q|C_j,d)}{P(q|d)} \cdot P(C_j|d) \cdot P(d|q).$$

We assume that $P(q|C_j, d) \approx P(q|d)$, that is, a probability of a query given a document can be determined without knowing the class of the query. This is the case for the majority of queries that are unambiguous. Counter examples are queries like 'jaguar' (animal and car brand) or 'apple' (fruit and computer manufacturer), but such ambiguous queries can not be classified *by definition*, and usually consists of common words. In this work we concentrate on rare queries, that tend to contain rare words, be longer, and match fewer documents; consequently in our setting this assumption mostly holds. Using this assumption, we can write $P(C_j|q) = \sum_{d \in D} P(C_j|d) \cdot P(d|q)$. The conditional probability of a classification for a given document $P(C_j|d)$ is estimated using the output of the document classifier (section 2.1). While $P(d|q)$ is harder to compute, we consider the underlying relevance model for ranking documents given a query. This issue is further explored in the next section.

[1] In the above examples, "SD450" and "nc4200" represent fairly old gadget models, and hence there are advertisers placing ads on these queries. However, in this paper we mainly deal with rare queries which are extremely difficult to match to relevant ads.

2.3 Classification-based relevance model

In order to describe a formal relationship of classification and ad placement (or search), we consider a model for using classification to determine ads (or search) relevance. Let a be an ad and q be a query, we denote by $R(a,q)$ the relevance of a to q. This number indicates how relevant the ad a is to query q, and can be used to rank ads a for a given query q. In this paper, we consider the following approximation of relevance function:

$$R(a,q) \approx R_C(a,q) = \sum_{C_j \in C} w(C_j) s(C_j,a) s(C_j,q). \quad (1)$$

The right hand-side expresses how we use the classification scheme C to rank ads, where $s(c,a)$ is a scoring function that specifies how likely a is in class c, and $s(c,q)$ is a scoring function that specifies how likely q is in class c. The value $w(c)$ is a weighting term for category c, indicating the importance of category c in the relevance formula.

This relevance function is an adaptation of the traditional word-based retrieval rules. For example, we may let categories be the words in the vocabulary. We take $s(C_j,a)$ as the word counts of C_j in a, $s(C_j,q)$ as the word counts of C_j in q, and $w(C_j)$ as the IDF term weighting for word C_j. With such choices, the method given by (1) becomes the standard TFIDF retrieval rule.

If we take $s(C_j,a) = P(C_j|a)$, $s(C_j,q) = P(C_j|q)$, and $w(C_j) = 1/P(C_j)$, and assume that q and a are independently generated given a hidden concept C, then we have

$$\begin{aligned} R_C(a,q) &= \sum_{C_j \in C} P(C_j|a) P(C_j|q)/P(C_j) \\ &= \sum_{C_j \in C} P(C_j|a) P(q|C_j)/P(q) = P(q|a)/P(q). \end{aligned}$$

That is, the ads are ranked according to $P(q|a)$. This relevance model has been employed in various statistical language modeling techniques for information retrieval. The intuition can be described as follows. We assume that a person searches an ad a by constructing a query q: the person first picks a concept C_j according to the weights $P(C_j|a)$, and then constructs a query q with probability $P(q|C_j)$ based on the concept C_j. For this query generation process, the ads can be ranked based on how likely the observed query is generated from each ad.

It should be mentioned that in our case, each query and ad can have multiple categories. For simplicity, we denote by C_j a random variable indicating whether q belongs to category C_j. We use $P(C_j|q)$ to denote the probability of q belonging to category C_j. Here the sum $\sum_{C_j \in C} P(C_j|q)$ may not equal to one. We then consider the following ranking formula:

$$R_C(a,q) = \sum_{C_j \in C} P(C_j|a) P(C_j|q). \quad (2)$$

We assume the estimation of $P(C_j|a)$ is based on an existing text-categorization system (which is known). Thus, we only need to obtain estimates of $P(C_j|q)$ for each query q.

Equation (2) is the ad relevance model that we consider in this paper, with unknown parameters $P(C_j|q)$ for each query q. In order to obtain their estimates, we use search results from major US search engines, where we assume that the ranking formula in (2) gives good ranking for search. That is, top results ranked by search engines should also be ranked high by this formula. Therefore given a query q, and top K result pages $d_1(q),\ldots,d_K(q)$ from a major search engine, we fit parameters $P(C_j|q)$ so that $R_C(d_i(q),q)$ have high scores for $i = 1,\ldots,K$. It is worth mentioning that using this method we can only compute relative strength of $P(C_j|q)$, but not the scale, because scale does not affect ranking. Moreover, it is possible that the parameters estimated may be of the form $g(P(C_j|q))$ for some monotone function $g(\cdot)$ of the actually conditional probability $g(P(C_j|q))$. Although this may change the meaning of the unknown parameters that we estimate, it does not affect the quality of using the formula to rank ads. Nor does it affect query classification with appropriately chosen thresholds. In what follows, we consider two methods to compute the classification information $P(C_j|q)$.

2.4 The voting method

We would like to compute $P(C_j|q)$ so that $R_C(d_i(q),q)$ are high for $i = 1,\ldots,K$ and $R_C(d,q)$ are low for a random document d. Assume that the vector $[P(C_j|d)]_{C_j \in C}$ is random for an average document, then the condition that $\sum_{C_j \in C} P(C_j|q)^2$ is small implies that $R_C(d,q)$ is also small averaged over d. Thus, a natural method is to maximize $\sum_{i=1}^K w_i R_C(d_i(q),q)$ subject to $\sum_{C_j \in C} P(C_j|q)^2$ being small, where w_i are weights associated with each rank i:

$$\max_{[P(\cdot|q)]} \left[\frac{1}{K} \sum_{i=1}^K w_i \sum_{C_j \in C} P(C_j|d_i(q)) P(C_j|q) - \lambda \sum_{C_j \in C} P(C_j|q)^2 \right],$$

where we assume $\sum_{i=1}^K w_i = 1$, and $\lambda > 0$ is a tuning regularization parameter. The optimal solution is

$$P(C_j|q) = \frac{1}{2\lambda} \sum_{i=1}^K w_i P(C_j|d_i(q)).$$

Since both $P(C_j|d_i(q))$ and $P(C_j|q)$ belong to $[0,1]$, we may just take $\lambda = 0.5$ to align the scale. In the experiment, we will simply take uniform weights w_i. A more complex strategy is to let w depend on d as well:

$$P(C_j|q) = \sum_d w(d,q) g(P(C_j|d)),$$

where $g(x)$ is a certain transformation of x.

In this general formulation, $w(d,q)$ may depend on factors other than the rank of d in the search engine results for q. For example, it may be a function of $r(d,q)$ where $r(d,q)$ is the relevance score returned by the underlying search engine. Moreover, if we are given a set of hand-labeled training category/query pairs (C,q), then both the weights $w(d,q)$ and the transformation $g(\cdot)$ can be learned using standard classification techniques.

2.5 Discriminative classification

We can treat the problem of estimating $P(C_j|q)$ as a classification problem, where for each q, we label $d_i(q)$ for $i = 1,\ldots,K$ as positive data, and the remaining documents as negative data. That is, we assign label $y_i(q) = 1$ for $d_i(q)$ when $i \le K$, and label $y_i(q) = -1$ for $d_i(q)$ when $i > K$.

In this setting, the classification scoring rule for a document $d_i(q)$ is linear. Let $x_i(q) = [P(C_j|d_i(q))]$, and $w = [P(C_j|q)]$, then $\sum_{C_j \in C} P(C_j|q) P(C_j|d_i(q)) = w \cdot x_i(q)$. The values $P(C_j|d)$ are the features for the linear classifier, and $[P(C_j|d)]$ is the weight vector, which can be computed using any linear classification method. In this paper, we consider estimating w using logistic regression [17] as follows: $P(\cdot|q) = \arg\min_w \sum_i \ln(1 + e^{-w \cdot x_i(q) y_i(q)})$.

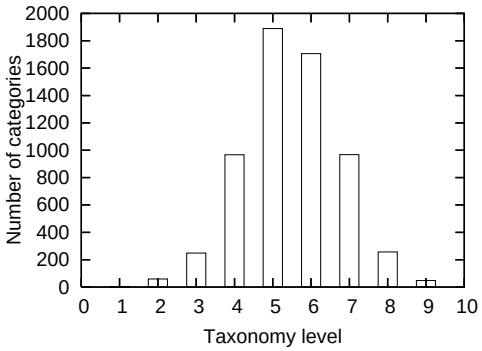

Figure 1: Number of categories by level

3. EVALUATION

In this section, we evaluate our methodology that uses Web search results for improving query classification.

3.1 Taxonomy

Our choice of taxonomy was guided by a Web advertising application. Since we want the classes to be useful for matching ads to queries, the taxonomy needs to be elaborate enough to facilitate ample classification specificity. For example, classifying all medical queries into one node will likely result in poor ad matching, as both "sore foot" and "flu" queries will end up in the same node. The ads appropriate for these two queries are, however, very different. To avoid such situations, the taxonomy needs to provide sufficient discrimination between common commercial topics. Therefore, in this paper we employ an elaborate taxonomy of approximately 6000 nodes, arranged in a hierarchy with median depth 5 and maximum depth 9. Figure 1 shows the distribution of categories by taxonomy levels. Human editors populated the taxonomy with labeled queries (approx. 150 queries per node), which were used as a training set; a small fraction of queries have been assigned to more than one category.

3.2 Digression: the basics of sponsored search

To discuss our set of evaluation queries, we need a brief introduction to some basic concepts of Web advertising. *Sponsored search* (or *paid search*) advertising is placing textual ads on the result pages of web search engines, with ads being driven by the originating query. All major search engines (Google, Yahoo!, and MSN) support such ads and act simultaneously as a search engine and an ad agency. These textual ads are characterized by one or more "bid phrases" representing those queries where the advertisers would like to have their ad displayed. (The name "bid phrase" comes from the fact that advertisers bid various amounts to secure their position in the tower of ads associated to a query. A discussion of bidding and placement mechanisms is beyond the scope of this paper [13].

However, many searches do not explicitly use phrases that someone bids on. Consequently, advertisers also buy "broad" matches, that is, they pay to place their advertisements on queries that constitute some modification of the desired bid phrase. In broad match, several syntactic modifications can be applied to the query to match it to the bid phrase, e.g., dropping or adding words, synonym substitution, etc. These transformations are based on rules and dictionaries. As advertisers tend to cover high-volume and high-revenue queries, broad-match queries fall into the tail of the distribution with respect to both volume and revenue.

3.3 Data sets

We used two representative sets of 1000 queries. Both sets contain queries that cannot be directly matched to advertisements, that is, none of the queries contains a bid phrase (this means we eliminated practically all popular queries).

The first set of queries can be matched to at least one ad using *broad* match as described above. Queries in the second set cannot be matched even by broad match, and therefore the search engine used in our study does not currently display any advertising for them. In a sense, these are even more rare queries and further away from common queries. As a measure of query rarity, we estimated their frequency in a month worth of query logs for a major US search engine; the median frequency was 1 for queries in Set 1 and 0 for queries in Set 2.

The queries in the two sets differ in their classification difficulty. In fact, queries in Set 2 are difficult to interpret even for human evaluators. Queries in Set 1 have on average 3.50 words, with the longest one having 11 words; queries in Set 2 have on average 4.39 words, with the longest query of 81 words. Recent studies estimate the average length of web queries to be just under 3 words[2], which is lower than in our test sets. As another measure of query difficulty, we measured the fraction of queries that contain quotation marks, as the latter assist query interpretation by meaningfully grouping the words. Only 8% queries in Set 1 and 14% in Set 2 contained quotation marks.

3.4 Methodology and evaluation metrics

The two sets of queries were classified into the target taxonomy using the techniques presented in section 2. Based on the confidence values assigned, the top 3 classes for each query were presented to human evaluators. These evaluators were trained editorial staff who possessed knowledge about the taxonomy. The editors considered every query-class pair, and rated them on the scale 1 to 4, with 1 meaning the classification is highly relevant and 4 meaning it is irrelevant for the query. About 2.4% queries in Set 1 and 5.4% queries in Set 2 were judged to be unclassifiable (e.g., random strings of characters), and were consequently excluded from evaluation. To compute evaluation metrics, we treated classifications with ratings 1 and 2 to be correct, and those with ratings 3 and 4 to be incorrect.

We used standard evaluation metrics: precision, recall and F1. In what follows, we plot precision-recall graphs for all the experiments. For comparison with other published studies, we also report precision and F1 values corresponding to complete recall ($R = 1$). Owing to the lack of space, we only show graphs for query Set 1; however, we show the numerical results for both sets in the tables.

3.5 Results

We compared our method to a baseline query classifier that does not use any external knowledge. Our baseline classifier expanded queries using standard query expansion techniques, grouped their terms using a phrase recognizer, boosted certain phrases in the query based on their statistical properties, and performed classification using the

[2]http://www.rankstat.com/html/en/seo-news1-most-people-use-2-word-phrases-in-search-engines.html

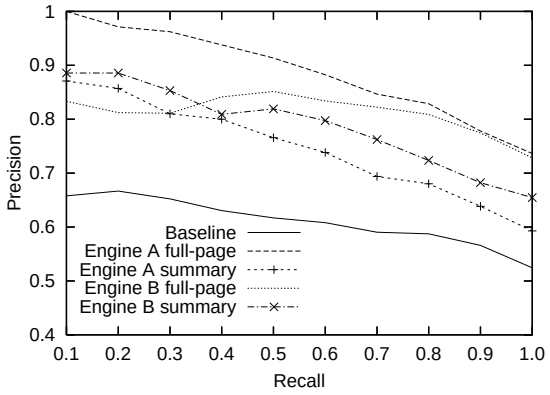

Figure 2: The effect of external knowledge

nearest-neighbor approach. This baseline classifier is actually a production version of the query classifier running in a major US search engine.

In our experiments, we varied values of pertinent parameters that characterize the exact way of using search results. In what follows, we start with the general assessment of the effect of using Web search results. We then proceed to exploring more refined techniques, such as using only search summaries versus actually crawling the returned URLs. We also experimented with using different numbers of search results per query, as well as with varying the number of classifications considered for each search result. For lack of space, we only show graphs for Set 1 queries and omit the graphs for Set 2 queries, which exhibit similar phenomena.

3.5.1 The effect of external knowledge

Queries by themselves are very short and difficult to classify. We use top search engine results for collecting background knowledge for queries. We employed two major US search engines, and used their results in two ways, either only summaries or the full text of crawled result pages. Figure 2 and Table 1 show that such extra knowledge considerably improves classification accuracy. Interestingly, we found that search engine A performs consistently better with full-page text, while search engine B performs better when summaries are used.

Engine	Context	Prec.	F1	Prec.	F1
		Set 1	Set 1	Set 2	Set 2
A	full-page	0.72	0.84	0.509	0.721
B	full-page	0.706	0.827	0.497	0.665
A	summary	0.586	0.744	0.396	0.572
B	summary	0.645	0.788	0.467	0.638
Baseline		0.534	0.696	0.365	0.536

Table 1: The effect of using external knowledge

3.5.2 Aggregation techniques

There are two major ways to use search results as additional knowledge. First, individual results can be classified separately, with subsequent voting among individual classifications. Alternatively, individual search results can be bundled together as one meta-document and classified as such using the document classifier. Figure 3 presents the results of these two approaches When full-text pages are used, the technique using individual classifications of search results evidently outperforms the bundling approach by a wide margin. However, in the case of summaries, bundling together is found to be consistently better than individual classification. This is because summaries by themselves are too short to be classified correctly individually, but when bundled together they are much more stable.

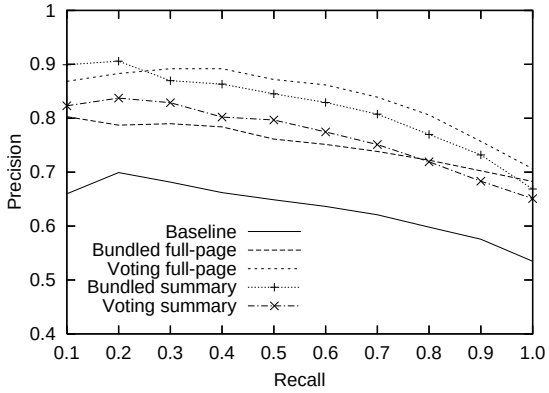

Figure 3: Voting vs. Bundling

3.5.3 Full page text vs. summary

To summarize the two preceding sections, background knowledge for each query is obtained by using either the full-page text or only the summaries of the top search results. Full page text was found to be more in conjunction with voted classification, while summaries were found to be useful when bundled together. The best results overall were obtained with full-page results classified individually, with subsequent voting used to determine the final query classification. This observation differs from findings by Shen et al. [20], who found summaries to be more useful. We attribute this distinction to the fact that the queries we used in this study are tail ones, which are rare and difficult to classify.

3.5.4 Varying the number of classes per search result

We also varied the number of classifications per search result, i.e., each result was permitted to have either 1, 3, or 5 classes. Figure 4 shows the corresponding precision-recall graphs for both full-page and summary-only settings. As can be readily seen, all three variants produce very similar results. However, the precision-recall curve for the 1-class experiment has higher fluctuations. Using 3 classes per search result yields a more stable curve, while with 5 classes per result the precision-recall curve is very smooth. Thus, as we increase the number of classes per result, we observe higher stability in query classification.

3.5.5 Varying the number of search results obtained

We also experimented with different numbers of search results per query. Figure 5 and Table 2 present the results of this experiment. In line with our intuition, we observed that classification accuracy steadily rises as we increase the number of search results used from 10 to 40, with a slight drop as we continue to use even more results (50). This is because using too few search results provides too little external knowledge, while using too many results introduces extra noise.

Using paired t-test, we assessed the statistical significance

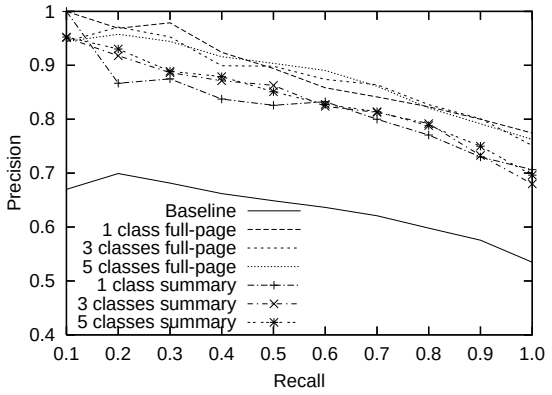

Figure 4: Varying the number of classes per page

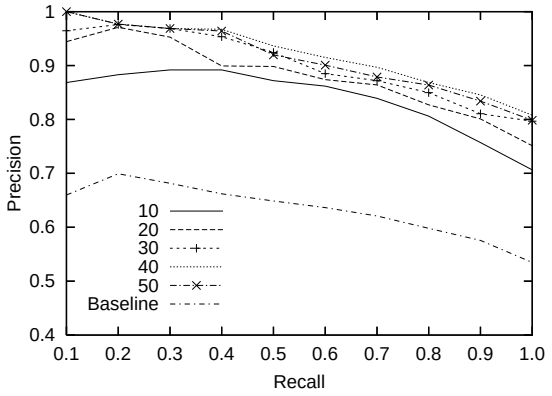

Figure 5: Varying the number of results per query

of the improvements due to our methodology versus the baseline. We found the results to be highly significant ($p < 0.0005$), thus confirming the value of external knowledge for query classification.

3.6 Voting versus alternative methods

As explained in Section 2.2, one may use several methods to classify queries from search engine results based on our relevance model. As we have seen, the voting method works quite well. In this section, we compare the performance of voting top-ten search results to the following two methods:

- A: Discriminative learning of query-classification based on logistic regression, described in Section 2.5.
- B: Learning weights based on quality score returned by a search engine. We discretize the quality score $s(d, q)$ of a query/document pair into {high, medium, low}, and learn the three weights w on a set of training queries, and test the performance on holdout queries. The classification formula, as explained at the end of Section 2.4, is $P(C_j|q) = \sum_d w(s(d, q))P(C_j|d)$.

Method B requires a training/testing split. Neither voting nor method A requires such a split; however, for consistency, we randomly draw 50-50 training/testing splits for ten times, and report the mean performance ± standard deviation on the test-split for all three methods. For this experiment, instead of precision and recall, we use DCG-k ($k = 1, 5$), popular in search engine evaluation. The DCG (discounted cumulated gain) metric, described in [8], is a ranking measure where the system is asked to rank a set of candidates (in our case, judged categories for each query), and computes for each query q: $DCG_k(q) = \sum_{i=1}^{k} g(C_i(q))/\log_2(i+1)$, where $C_i(q)$ is the i-th category for query q ranked by the system, and $g(C_i)$ is the grade of C_i: we assign grade of $10, 5, 1, 0$ to the 4-point judgment scale described earlier to compute DCG. The decaying choice of $\log_2(i+1)$ is conventional, which does not have particular importance. The overall DCG of a system is the averaged DCG over queries. We use this metric instead of precision/recall in this experiment because it can directly handle multi-grade output. Therefore as a single metric, it is convenient for comparing the methods. Note that precision/recall curves used in the earlier sections yield some additional insights not immediately apparent from the DCG numbers.

Number of results	Precision	F1
baseline	0.534	0.696
10	0.706	0.827
20	0.751	0.857
30	0.796	0.886
40	**0.807**	**0.893**
50	0.798	0.887

Table 2: Varying the number of search results

Set 1		
Method	DCG-1	DCG-5
Oracle	7.58 ± 0.19	14.52 ± 0.40
Voting	5.28 ± 0.15	11.80 ± 0.31
Method A	5.48 ± 0.16	12.22 ± 0.34
Method B	5.36 ± 0.18	12.15 ± 0.35
Set 2		
Method	DCG-1	DCG-5
Oracle	5.69 ± 0.18	9.94 ± 0.32
Voting	3.50 ± 0.17	7.80 ± 0.28
Method A	3.63 ± 0.23	8.11 ± 0.33
Method B	3.55 ± 0.18	7.99 ± 0.31

Table 3: Voting and alternative methods

Results from our experiments are given in Table 3. The oracle method is the best ranking of categories for each query after seeing human judgments. It cannot be achieved by any realistic algorithm, but is included here as an absolute upper bound on DCG performance. The simple voting method performs very well in our experiments. The more complicated methods may lead to moderate performance gain (especially method A, which uses discriminative training in Section 2.5). However, both methods are computationally more costly, and the potential gain is minor enough to be neglected. This means that as a simple method, voting is quite effective.

We can observe that method B, which uses quality score returned by a search engine to adjust importance weights of returned pages for a query, does not yield appreciable improvement. This implies that putting equal weights (voting) performs similarly as putting higher weights to higher quality documents and lower weights to lower quality documents (method B), at least for the top search results. It may be possible to improve this method by including other page-features that can differentiate top-ranked search results. However, the effectiveness will require further inves-

tigation which we did not test. We may also observe that the performance on Set 2 is lower than that on Set 1, which means queries in Set 2 are harder than those in Set 1.

3.7 Failure analysis

We scrutinized the cases when external knowledge did not improve query classification, and identified three main causes for such lack of improvement. (1)Queries containing random strings, such as telephone numbers — these queries do not yield coherent search results, and so the latter cannot help classification (around 5% of queries were of this kind). (2) Queries that yield no search results at all; there were 8% such queries in Set 1 and 15% in Set 2. (3) Queries corresponding to recent events, for which the search engine did not yet have ample coverage (around 5% of queries). One notable example of such queries are entire names of news articles—if the exact article has not yet been indexed by the search engine, search results are likely to be of little use.

4. RELATED WORK

Even though the average length of search queries is steadily increasing over time, a typical query is still shorter than 3 words. Consequently, many researchers studied possible ways to enhance queries with additional information.

One important direction in enhancing queries is through query expansion. This can be done either using electronic dictionaries and thesauri [22], or via relevance feedback techniques that make use of a few top-scoring search results. Early work in information retrieval concentrated on manually reviewing the returned results [16, 15]. However, the sheer volume of queries nowadays does not lend itself to manual supervision, and hence subsequent works focused on *blind* relevance feedback, which basically assumes top returned results to be relevant [23, 12, 4, 14].

More recently, studies in query augmentation focused on classification of queries, assuming such classifications to be beneficial for more focused query interpretation. Indeed, Kowalczyk et al. [10] found that using query classes improved the performance of document retrieval.

Studies in the field pursue different approaches for obtaining additional information about the queries. Beitzel et al. [1] used semi-supervised learning as well as unlabeled data [2]. Gravano et al. [6] classified queries with respect to geographic locality in order to determine whether their intent is local or global.

The 2005 KDD Cup on web query classification inspired yet another line of research, which focused on enriching queries using Web search engines and directories [11, 18, 20, 9, 21]. The KDD task specification provided a small taxonomy (67 nodes) along with a set of labeled queries, and posed a challenge to use this training data to build a query classifier. Several teams used the Web to enrich the queries and provide more context for classification. The main research questions of this approach the are (1) how to build a document classifier, (2) how to translate its classifications into the target taxonomy, and (3) how to determine the query class based on document classifications.

The winning solution of the KDD Cup [18] proposed using an ensemble of classifiers in conjunction with searching multiple search engines. To address issue (1) above, their solution used the Open Directory Project (ODP) to produce an ODP-based document classifier. The ODP hierarchy was then mapped into the target taxonomy using word matches at individual nodes. A document classifier was built for the target taxonomy by using the pages in the ODP taxonomy that appear in the nodes mapped to the particular target node. Thus, Web documents were first classified with respect to the ODP hierarchy, and their classifications were subsequently mapped to the target taxonomy for query classification.

Compared to this approach, we solved the problem of document classification directly in the target taxonomy by using the queries to produce document classifier as described in Section 2. This simplifies the process and removes the need for mapping between taxonomies. This also streamlines taxonomy maintenance and development. Using this approach, we were able to achieve good performance in a very large scale taxonomy. We also evaluated a few alternatives how to combine individual document classifications when actually classifying the query.

In a follow-up paper [19], Shen et al. proposed a framework for query classification based on bridging between two taxonomies. In this approach, the problem of not having a document classifier for web results is solved by using a training set available for documents with a different taxonomy. For this, an intermediate taxonomy with a training set (ODP) is used. Then several schemes are tried that establish a correspondence between the taxonomies or allow for mapping of the training set from the intermediate taxonomy to the target taxonomy. As opposed to this, we built a document classifier for the target taxonomy directly, without using documents from an intermediate taxonomy. While we were not able to directly compare the results due to the use of different taxonomies (we used a much larger taxonomy), our precision and recall results are consistently higher even over the hardest query set.

5. CONCLUSIONS

Query classification is an important information retrieval task. Accurate classification of search queries is likely to benefit a number of higher-level tasks such as Web search and ad matching. Since search queries are usually short, by themselves they usually carry insufficient information for adequate classification accuracy. To address this problem, we proposed a methodology for using search results as a source of external knowledge. To this end, we send the query to a search engine, and assume that a plurality of the highest-ranking search results are relevant to the query. Classifying these results then allows us to classify the original query with substantially higher accuracy.

The results of our empirical evaluation definitively confirmed that using the Web as a repository of world knowledge contributes valuable information about the query, and aids in its correct classification. Notably, our method exhibits significantly higher accuracy than methods described in prior studies[3] Compared to prior studies, our approach does not require any auxiliary taxonomy, and we produce a query classifier directly for the target taxonomy. Furthermore, the taxonomy used in this study is approximately 2 orders of magnitude larger than that used in prior works.

We also experimented with different values of parameters that characterize our method. When using search results, one can either use only summaries of the results provided by

[3]Since the field of query classification does not yet have established and agreed upon benchmarks, direct comparison of results is admittedly tricky.

the search engine, or actually crawl the results pages for even deeper knowledge. Overall, query classification performance was the best when using the full crawled pages (Table 1). These results are consistent with prior studies [5], which found that using full crawled pages is superior for document classification than using only brief summaries. Our findings, however, are different from those reported by Shen et al. [19], who found summaries to yield better results. We attribute our observations to using a more elaborate voting scheme among the classifications of individual search results, as well as to using a more difficult set of rare queries.

In this study we used two major search engines, A and B. Interestingly, we found notable distinctions in the quality of their output. Notably, for engine A the overall results were better when using the full crawled pages of the search results, while for engine B it seems to be more beneficial to use the summaries of results. This implies that while the quality of search results returned by engine A is apparently better, engine B does a better work in summarizing the pages.

We also found that the best results were obtained by using full crawled pages and performing voting among their individual classifications. For a classifier that is external to the search engine, retrieving full pages may be prohibitively costly, in which case one might prefer to use summaries to gain computational efficiency. On the other hand, for the owners of a search engine, full page classification is much more efficient, since it is easy to preprocess all indexed pages by classifying them once onto the (fixed) taxonomy. Then, page classifications are obtained as part of the meta-data associated with each search result, and query classification can be nearly instantaneous.

When using summaries it appears that better results are obtained by first concatenating individual summaries into a meta-document, and then using its classification as a whole. We believe the reason for this observation is that summaries are short and inherently noisier, and hence their aggregation helps to correctly identify the main theme. Consistent with our intuition, using too few search results yields useful but insufficient knowledge, and using too many search results leads to inclusion of marginally relevant Web pages. The best results were obtained when using 40 top search hits.

In this work, we first classify search results, and then use their classifications directly to classify the original query. Alternatively, one can use the classifications of search results as *features* in order to learn a second-level classifier. In Section 3.6, we did some preliminary experiments in this direction, and found that learning such a secondary classifier did not yield considerably advantages. We plan to further investigate this direction in our future work.

It is also essential to note that implementing our methodology incurs little overhead. If the search engine classifies crawled pages during indexing, then at query time we only need to fetch these classifications and do the voting.

To conclude, we believe our methodology for using Web search results holds considerable promise for substantially improving the accuracy of Web search queries. This is particularly important for rare queries, for which little per-query learning can be done, and in this study we proved that such scarceness of information could be addressed by leveraging the knowledge found on the Web. We believe our findings will have immediate applications to improving the handling of rare queries, both for improving the search results as well as yielding better matched advertisements. In our further research we also plan to make use of session information in order to leverage knowledge about previous queries to better classify subsequent ones.

6. REFERENCES

[1] S. Beitzel, E. Jensen, O. Frieder, D. Grossman, D. Lewis, A. Chowdhury, and A. Kolcz. Automatic web query classification using labeled and unlabeled training data. In *Proceedings of SIGIR'05*, 2005.

[2] S. Beitzel, E. Jensen, O. Frieder, D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of ICDM'05*, 2005.

[3] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[4] E. Efthimiadis and P. Biron. UCLA-Okapi at TREC-2: Query expansion experiments. In *TREC-2*, 1994.

[5] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI'05*, pages 1048–1053, 2005.

[6] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. Categorizing web queries according to geographical locality. In *CIKM'03*, 2003.

[7] E. Han and G. Karypis. Centroid-based document classification: Analysis and experimental results. In *PKDD'00*, September 2000.

[8] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR'00*, 2000.

[9] Z. Kardkovacs, D. Tikk, and Z. Bansaghi. The ferrety algorithm for the KDD Cup 2005 problem. In *SIGKDD Explorations*, volume 7. ACM, 2005.

[10] P. Kowalczyk, I. Zukerman, and M. Niemann. Analyzing the effect of query class on document retrieval performance. In *Proc. Australian Conf. on AI*, pages 550–561, 2004.

[11] Y. Li, Z. Zheng, and H. Dai. KDD CUP-2005 report: Facing a great challenge. In *SIGKDD Explorations*, volume 7, pages 91–99. ACM, December 2005.

[12] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *SIGIR'98*, pages 206–214, 1998.

[13] M. Moran and B. Hunt. *Search Engine Marketing, Inc.: Driving Search Traffic to Your Company's Web Site*. Prentice Hall, Upper Saddle River, NJ, 2005.

[14] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC-3*, 1995.

[15] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[16] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *JASIS*, 41(4):288–297, 1990.

[17] T. Santner and D. Duffy. *The Statistical Analysis of Discrete Data*. Springer-Verlag, 1989.

[18] D. Shen, R. Pan, J. Sun, J. Pan, K. Wu, J. Yin, and Q. Yang. Q2C@UST: Our winning solution to query classification in KDDCUP 2005. In *SIGKDD Explorations*, volume 7, pages 100–110. ACM, 2005.

[19] D. Shen, R. Pan, J. Sun, J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM TOIS*, 24:320–352, July 2006.

[20] D. Shen, J. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *SIGIR'06*, pages 131–138, 2006.

[21] D. Vogel, S. Bickel, P. Haider, R. Schimpfky, P. Siemen, S. Bridges, and T. Scheffer. Classifying search engine queries using the web as background knowledge. In *SIGKDD Explorations*, volume 7. ACM, 2005.

[22] E. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR'94*, 1994.

[23] J. Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM TOIS*, 18(1):79–112, 2000.