

Leveraging Search Engine Results for Query Classification*

Shruti K. Bhandari and Brian D. Davison
Department of Computer Science & Engineering
Lehigh University
email: {skb206,davison}@cse.lehigh.edu.

Abstract

Web query classification is significant to search engines for the purpose of efficient retrieval of appropriate results in response to user queries. User queries are short in nature, contain noise and are ambiguous in terms of user intent. In this paper, we present different features—such as snippets, page content and titles of search engine results for a given query—that can be used to enhance user queries for classification purposes. We train various classifiers using different features so that these classifiers can be evaluated on a set of test queries. We show that the performance of our technique of classification of a query using the snippets of search engine results for that query is comparable to that obtained by the solutions provided by the winning teams at the KDD Cup 2005 competition, in spite of the fact that our technique is less complex in comparison to the other winning solutions. It also noticeably outperforms query classification using query text directly or the content of pages returned by search engines for the query.

1 INTRODUCTION

With the exponential increase in the amount of information available on the Internet, the number of people searching for and using this information has been growing at the same rate. The search engines have become a necessity for users who wish to find different kinds of information varying from research articles to shopping guides and from news to pictures of their favorite actors. Although the current search engines provide very good results to users in response to the search queries, the effectiveness of these systems can be increased by topical classification of user queries.

Current search engines may direct the incoming user query to a set of topic-specific databases and then provide the combined result list to the user. By incorporating a good topical query classification method prior to directing the query to these databases, the computational cost can be reduced by directing the query to an appropriate and a smaller concentrated set of databases. Also, by using query classification, search engines can return different set of results for different users (for example, personalized search). Query classification can also be used to help online advertisement services to place appropriate ads according to the topic of the query being searched [BFG⁺07].

The problem of query classification is different from traditional document classification in the sense that queries are typically quite short. They do not provide sufficient features that can be extracted for supervised learning and classification. They contain a lot of noise due to spelling mistakes or other lexical errors made by users while entering the queries. The intent of the queries is also not clear as different users may

*Technical Report LU-CSE-07-013, Dept. of Computer Science and Engineering, Lehigh University, Bethlehem, PA, 18015.

tend to search for different kinds of information using the same query and use different terms for the same information need.

As a result of these issues, text based classifiers such as Rainbow and SVM are ineffective in classifying web user queries. In our experiments, we intend to use these classifiers in the traditional fashion but will enhance or replace the queries with additional text to provide more high-quality features. Instead of classifying the query text directly, we train the classifier using features obtained by submitting the queries to a search engine and then using the snippets, titles, URLs and page contents of the returned results in a variety of combinations as input documents to the classifier.

In this paper, we have tried to answer the following questions: 1) Is there a technique less complex than the techniques submitted at the KDD Cup 2005 competition but gives comparable performance to theirs? 2) Does enhancing the queries with related text help in improving query classification? and 3) What combination of features, used for enhancing the queries, gives the best performance?

After tuning the classifier using n-fold cross-validation on the 111 training data queries, we tested it on the 800 queries used to evaluate the solutions submitted at the KDD Cup 2005 competition. The results obtained are comparable with those obtained by the winning teams of the competition but with a much simpler approach.

The remainder of the paper is organized as follows: we first summarize related work carried out in web query classification in Section 2. In Section 3, we explain the classification approach that we followed. The data sets used and the experimental results are discussed and analyzed in Section 4. Finally, we summarize in Section 5.

2 Related Work

The problem of query classification received significant attention as the focus of the 2005 KDD Cup competition [LZ05]. The task of competition was to classify 800,000 web user search queries into 67 predefined categories. The participants were provided with a small set of 111 queries with labeled categories as a sample of the mapping between the query strings and the categories. The participants were required to categorize each of the 800,000 queries into up to five categories. The evaluation process [LZD05] was as follows:

$$\text{Precision} = \frac{\sum_i \text{number of queries correctly tagged as } c_i}{\sum_i \text{number of queries tagged as } c_i} \quad (1)$$

$$\text{Recall} = \frac{\sum_i \text{number of queries correctly tagged as } c_i}{\sum_i \text{number of queries whose category is labeled as } c_i \text{ by human labeler}} \quad (2)$$

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

A random set of 800 queries was selected from the 800,000 for evaluation purposes.

There were three awards, one each for precision, performance, and creativity. The top three F_1 scores achieved by the teams considered for the Performance Award were 0.444, 0.405 and 0.384, and the top three precision values achieved by the teams considered for the Precision Award were 0.424, 0.341 and 0.340 [LZD05].

The techniques used for query classification in this competition include 1) Mapping pre-defined/existing directory structure to KDD Cup categories and 2) Constructing the mappings between KDD Cup categories and words (descriptions), and then using the mappings to answer the categories of search queries that were treated as a bag of words. Most participants adopted machine learning algorithms like Naive Bayesian

classifier, SVM, KNN, Neural Network, or Logistic Regression in their solutions. A few participants constructed multiple models and then combined them together to achieve better results. Some of them used distance/probability as criteria to combine predictions. Some applied ad hoc rules in combining predictions. Others adopted ensemble learning (e.g., Boosting). More advanced methods were adopted to tune model parameters, such as using manually tagged examples to tune model parameters or using reward/penalty factors in model tuning/training. Other learning approaches taken by participants were transforming multi-class problem into multiple binary-class problems and iterative learning.

Two of the award-winning approaches are worth mentioning, as they both incorporate some form of intermediate structure. Shen et al. [SPS⁺06] presented a technique called query enrichment, which takes a short query and maps it to intermediate objects. Kardkovacs et al. [KTB05] presented the Ferrety algorithm which can be generalized for mapping one taxonomy to another if training documents are available.

Additional advances have been made. Beitzel et al. [BJF⁺05] applied computational linguistics to mine the unlabeled data in web query logs to improve automated query classification. More recently, Beitzel et al. [BJCF07] examined pre- versus post-retrieval classification effectiveness and the effect of training explicitly from classified queries versus bridging a classifier trained using a document taxonomy. Shen et al. [SSYC06] described another novel approach for query classification which outperforms the winning solution of the KDD Cup 2005 competition. They first built a bridging classifier on an intermediate taxonomy in an offline mode. This classifier is then used online to map user queries to target categories via the intermediate taxonomy. Broder et al. [BFG⁺07] looked at the problem of classification of rare queries, and like our approach below, used query search results to build pseudo-documents for classification.

3 Our Approach

In this section, we discuss the different features and methods used in our work.

We used the set of 111 labeled queries provided by the KDD Cup 2005 competition dataset for training and testing initially for cross-validation purposes. The cross-validation or ‘leave one out’ classification method would help determine which features are more useful so that they could then be used to evaluate any given set of test queries. Evaluation was via the F_1 measure similar to the method used in the KDD Cup 2005 competition. We classified the queries into the 67 categories used at the competition.

Initially, we employed the Rainbow text based classifier [McC96], using the Naive Bayes algorithm, to classify the queries by using only the query terms. The results, as presented in the next section, indicated that the F_1 (harmonic mean of precision and recall) score, thus obtained, was 0.323 which is very low as compared to those obtained by the winning solutions at KDD Cup 2005. We then tried various combinations of features in different ways to achieve better results.

Given the 111 labeled queries, we submitted each to three search engines (Google, Microsoft and Yahoo) and obtained the top 50 results from each. We obtained the snippets (short query- and engine-specific descriptions of the result pages), titles, URLs, and page contents for each of the 50 results for each search engine (thus collecting a total of 150 result sets per query).

3.1 Preprocessing

Prior to using the snippets, titles, URLs and page contents, they were processed by removing HTML tags and HTML-specific content such as ``, `<p>`, `&` and `"`. Non-alphanumeric characters, such as colons, were also removed from URLs. These were then replaced by whitespace.

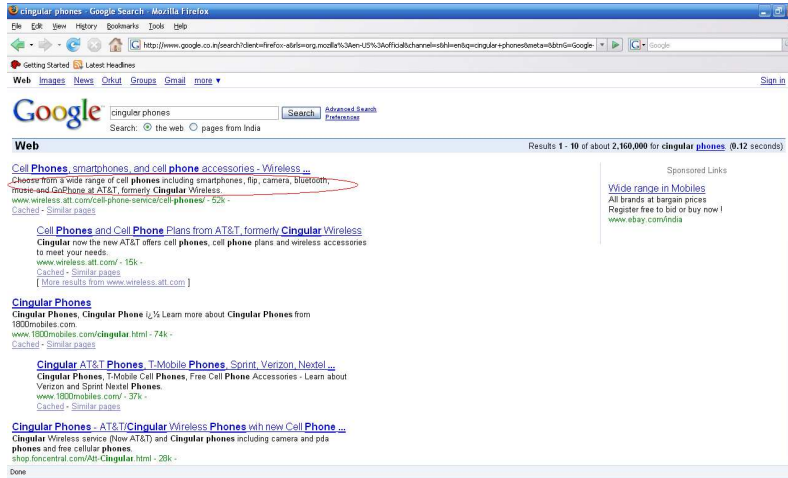


Figure 1: Search Engine Results.

This preprocessing was required as the HTML markup and non-alphanumeric characters do not contribute in the classification process. In addition, HTML tags would likely add noise to the classification process, reducing performance.

3.2 Document Generation

The input documents to the Rainbow classifier contained the snippets, titles, page contents and URLs in a variety of combinations. Consider Figure 1 in which the search engine results for the query ‘cingular phones’ are displayed. An example of the snippet used in the input document for the classification purpose is marked in red. The hyperlinked text in each search result indicates the title of the result page. These titles and snippets of the top 50 results are used in the input documents for classification of the query ‘cingular phones’. Similarly, the URLs of the top 50 results and the contents of those pages are also used in the input documents.

In the rest of this subsection we describe the different approaches we used to build a document for classification that would represent the query.

3.2.1 Single document per query-specific search engine results

In the first method, we trained and tested the query classifier using artificial documents, each containing snippets of the top 50 search results returned by Google for the query to be classified. Similarly, titles and page contents of the top 50 Google search results were used as features for classification of each query. The above three tests were carried out using Microsoft and Yahoo! search results in a similar manner.

3.2.2 Combined results from all three search engines

We combined the 50 snippets each of all three search engines to have a single document for each query as another feature for classification. Similarly, combined 150 titles and 150 page contents were used as two other features.

3.2.3 Combination of features

We tried the following combinations of features using combined 50 results of each search engine and also, the combined 150 results of all three search engines, as a single document per query.

- URL + snippet + title
- snippet + title
- snippet + URL

Training and testing the classifier using only page content from result URLs gave lower F_1 scores as shown in Section 4. This suggests that the large amount of information in the page content introduced noise. Hence, page content was not further combined with any other feature.

3.2.4 Top 10 snippets

The F_1 scores obtained using snippets were high using the 50 results from each search engine as well as the combined 150 results for each query. If the large number of terms in page content introduced noise, there was a possibility that 50 snippets from each search engine combined into a single document would do the same. Hence, we tried using only the top 10 snippets from each search engine and the combined 30 results as two other features in our classification work.

3.2.5 Separate vs. combined search engine results

The results obtained by submitting the queries to the three search engines were used in two different ways for the purpose of classification.

In the first set of experiments, the 50 snippets (titles, page contents and other combination of the features) from each search engine were combined into a single document for each query while training and testing the queries resulting in a set of 111 documents to build the training model. The same was repeated by combining all 150 results into a single document for each query. We shall call this set of features as macro-level.

In the second set of experiments, each of the 50 snippets (titles and other combination of features) from each search engine was treated as a separate document for training and testing purposes. The same was repeated by treating each of the 150 results from all three search engines as a single document for each query. We call this set of features as micro-level.

3.3 Hard vs. Soft Classification

For each of the features described above, we used two different ways of evaluating the results of the classifier.

In the first method, hard classification, we considered only the top class returned by the classifier for each test query for evaluation. Thus, each query was classified into 1 class each. The precision, recall and F_1 measures were thus calculated with the total number of 111 classes into which the 111 queries were classified. Thus, the denominator for the calculation of precision in Equation 1 is constant (111).

The second method, soft classification, involves the use of the top five classes (if present) returned by the classifier for evaluation purposes. If the classifier returns less than five classes, only those classes are used for evaluation. Thus, for the 111 tests performed, the denominator for the calculation of precision can vary from 111 (classifier returns one class for each test query) to 555 (classifier returns five classes or more for each test query). For example, if the classifier returns a score greater than zero for five or more classes, we consider each of the top five classes to be the classification for the given query. Thus, for 111 queries, we have a total of 555 classifications. Hence, the denominator in Equation 1 is 555.

4 Experiments and Evaluation Results

4.1 Dataset Used

The 111 labeled queries from the KDD Cup 2005 competition were used to train and tune the classifier using various features such as snippets, titles etc. In the KDD Cup 2005 competition, the set of 111 queries were labeled to demonstrate the mapping between the query strings and the 67 categories. The participants of the competition could use this set of queries in the way they wished. We used the given set to train the classifier and then to test the classifier to find the best set of features which could then be used to test any set of test queries.

4.2 Experiments and Evaluation

4.2.1 Rainbow classifier

The model for the Rainbow classifier [McC96] was built using the snippets (and other features described in Section 3, one at a time) as input documents. The classifier was then tested 111 times for each feature by placing 110 documents as the training documents and the remaining one as the test document ('leave one out classification' or 'n-1 cross-validation'). The results for these 111 tests were then used to find the set of features producing the best results. In each of the 111 tests, the query was classified into a maximum of five categories in case of soft classification using the macro-level feature set method described in Section 3.2.5. In case of using micro-level feature set, the number of documents for each query was not 1 but up to 50 (in case of using each search engine separately) or up to 150 (for the combination of all three search engine results). Hence, for our 'leave one out' classification, all of the documents for 110 queries were used for training and all of the documents for the remaining one query were used as test documents once each. The final category of the test query was determined by calculating the average of the classification scores of all classification results obtained using all documents belonging to that query. When the search results (snippets, titles, URLs and page contents) from the three search engines were combined as a feature, duplicates, if present, across the search engine results were retained as they provide reinforcement for the results on which the search engines agree.

The first test of the 111 queries was based on the use of only the query terms as the feature. The results based on this feature using soft and hard classification are shown in Table 1.

| | Precision | Recall | F_1 |
|---------------------|-----------|--------|--------------|
| Soft Classification | 0.285 | 0.374 | 0.323 |
| Hard Classification | 0.432 | 0.114 | 0.18 |

Table 1: Rainbow-based classification using query terms only.

Tables 2 to 5 show the results obtained by using the search results of MSN, Google and Yahoo! using soft and hard classification.

| | MSN | | | Google | | | Yahoo | | |
|-------------------------------|-------|--------|--------------|--------|--------|--------------|-------|--------|--------------|
| | Prec. | Recall | F_1 | Prec. | Recall | F_1 | Prec. | Recall | F_1 |
| Snippet | 0.482 | 0.353 | 0.408 | 0.438 | 0.348 | 0.388 | 0.466 | 0.353 | 0.402 |
| Page Content | 0.643 | 0.175 | 0.275 | 0.661 | 0.18 | 0.283 | 0.702 | 0.201 | 0.312 |
| URL+Snippet+Title | 0.587 | 0.289 | 0.388 | 0.555 | 0.275 | 0.367 | 0.592 | 0.289 | 0.388 |
| Snippet+Title | 0.483 | 0.31 | 0.378 | 0.467 | 0.301 | 0.366 | 0.512 | 0.308 | 0.385 |
| Snippet+URL | 0.584 | 0.313 | 0.408 | 0.549 | 0.32 | 0.404 | 0.545 | 0.315 | 0.399 |
| Title only | 0.372 | 0.408 | 0.389 | 0.411 | 0.448 | 0.429 | 0.404 | 0.531 | 0.459 |
| Snippets (top 10) | 0.367 | 0.476 | 0.414 | 0.362 | 0.472 | 0.41 | 0.382 | 0.5 | 0.433 |
| Sep. docs (snippet) | 0.416 | 0.547 | 0.472 | 0.402 | 0.528 | 0.457 | 0.423 | 0.557 | 0.481 |
| Sep. docs (snippet+URL+title) | 0.404 | 0.531 | 0.459 | 0.393 | 0.517 | 0.447 | 0.413 | 0.543 | 0.469 |
| Sep. docs (snippet+URL) | 0.402 | 0.528 | 0.456 | 0.404 | 0.531 | 0.459 | 0.405 | 0.533 | 0.46 |
| Sep. docs (snippet+title) | 0.414 | 0.545 | 0.471 | 0.391 | 0.514 | 0.444 | 0.42 | 0.552 | 0.477 |
| Sep. docs (title only) | 0.377 | 0.495 | 0.428 | 0.378 | 0.498 | 0.43 | 0.373 | 0.491 | 0.424 |
| Sep. docs (top 10 snippets) | 0.382 | 0.502 | 0.434 | 0.369 | 0.486 | 0.419 | 0.382 | 0.502 | 0.434 |

Table 2: Separate soft classification results using Rainbow on training set.

| Combined (MSN+Google+Yahoo) | Precision | Recall | F_1 |
|-----------------------------------|-----------|--------|--------------|
| Snippet | 0.622 | 0.242 | 0.348 |
| Page Content | 0.713 | 0.194 | 0.305 |
| URL+Snippet+Title | 0.667 | 0.223 | 0.334 |
| Snippet+Title | 0.601 | 0.218 | 0.32 |
| Snippet+URL | 0.674 | 0.225 | 0.337 |
| Title only | 0.463 | 0.31 | 0.371 |
| Snippets (top 10) | 0.431 | 0.431 | 0.431 |
| Separate docs (snippet) | 0.432 | 0.569 | 0.492 |
| Separate docs (snippet+URL+title) | 0.431 | 0.566 | 0.489 |
| Separate docs (snippet+URL) | 0.42 | 0.552 | 0.477 |
| Separate docs (snippet+title) | 0.42 | 0.552 | 0.477 |
| Separate docs (title only) | 0.405 | 0.533 | 0.46 |
| Separate docs (top 10 snippets) | 0.405 | 0.533 | 0.46 |

Table 3: Combined soft classification results using Rainbow on training set.

| | MSN | | | Google | | | Yahoo | | |
|-----------------------------------|-------|--------|--------------|--------|--------|--------------|-------|--------|--------------|
| | Prec. | Recall | F_1 | Prec. | Recall | F_1 | Prec. | Recall | F_1 |
| Snippet | 0.685 | 0.18 | 0.285 | 0.613 | 0.161 | 0.255 | 0.667 | 0.175 | 0.277 |
| Page Content | 0.64 | 0.168 | 0.266 | 0.658 | 0.173 | 0.274 | 0.703 | 0.185 | 0.293 |
| URL+Snippet+Title | 0.667 | 0.175 | 0.277 | 0.685 | 0.18 | 0.285 | 0.712 | 0.187 | 0.296 |
| Snippet+Title | 0.685 | 0.18 | 0.285 | 0.613 | 0.161 | 0.255 | 0.694 | 0.182 | 0.288 |
| Snippet+URL | 0.685 | 0.18 | 0.285 | 0.73 | 0.192 | 0.304 | 0.676 | 0.178 | 0.282 |
| Title only | 0.559 | 0.147 | 0.233 | 0.613 | 0.161 | 0.255 | 0.64 | 0.168 | 0.266 |
| Snippets (top 10) | 0.613 | 0.161 | 0.255 | 0.64 | 0.168 | 0.266 | 0.604 | 0.159 | 0.252 |
| Sep. docs (snippet) | 0.703 | 0.185 | 0.293 | 0.703 | 0.185 | 0.293 | 0.667 | 0.175 | 0.277 |
| Sep. docs (snippet + URL + title) | 0.667 | 0.175 | 0.277 | 0.649 | 0.171 | 0.271 | 0.721 | 0.189 | 0.299 |
| Sep. docs (snippet + URL) | 0.694 | 0.182 | 0.288 | 0.685 | 0.18 | 0.285 | 0.694 | 0.182 | 0.288 |
| Sep. docs (snippet + title) | 0.676 | 0.178 | 0.282 | 0.658 | 0.173 | 0.274 | 0.703 | 0.185 | 0.293 |
| Sep. docs (title only) | 0.586 | 0.154 | 0.244 | 0.64 | 0.168 | 0.266 | 0.667 | 0.175 | 0.277 |
| Sep. docs (top 10 snippets) | 0.649 | 0.171 | 0.271 | 0.667 | 0.175 | 0.277 | 0.64 | 0.168 | 0.266 |

Table 4: Separate hard classification results using Rainbow on training set.

| Combined (MSN+Google+Yahoo) | Precision | Recall | F_1 |
|---------------------------------------|-----------|--------|--------------|
| Snippet | 0.694 | 0.182 | 0.288 |
| Page Content | 0.712 | 0.187 | 0.296 |
| URL+Snippet+Title | 0.694 | 0.182 | 0.288 |
| Snippet+Title | 0.676 | 0.178 | 0.282 |
| Snippet+URL | 0.685 | 0.18 | 0.285 |
| Title only | 0.658 | 0.173 | 0.274 |
| Snippets (top 10) | 0.64 | 0.168 | 0.266 |
| Separate docs (snippet) | 0.685 | 0.18 | 0.285 |
| Separate docs (snippet + URL + title) | 0.676 | 0.178 | 0.282 |
| Separate docs (snippet + URL) | 0.703 | 0.185 | 0.293 |
| Separate docs (snippet + title) | 0.694 | 0.182 | 0.289 |
| Separate docs (title only) | 0.622 | 0.164 | 0.26 |
| Separate docs (top 10 snippets) | 0.676 | 0.178 | 0.282 |

Table 5: Combined hard classification results using Rainbow on training set.

We then evaluated the performance of this classifier by testing it on the 800 queries used to judge the submitted solutions at the KDD Cup 2005 competition. Observing the results obtained by the cross-validation tests, we decided to evaluate the 800 queries using only the top 50 and the top 10 snippets as the features for query enhancement as these features consistently gave good results in the cross-validation method.

In accordance with the evaluation method used in the KDD Cup 2005 competition, we tested the 800 queries against the manually labeled queries of all three labelers. The results thus obtained were then averaged to obtain the final results.

Tables 6 and 7 show the evaluation results after testing the classifier on the 800 test queries. The evaluation results show that the best performance values, achieved by using the top 50 snippets of the Google search results and the top 50 snippets using all three search engine results combined, are comparable to the top three submitted solutions at the KDD Cup 2005 competition. However, the technique suggested in this work is considerably simpler than the top submitted solutions.

| | | Judge 1 | Judge 2 | Judge 3 | Overall |
|--|-----------|---------|---------|---------|--------------|
| Top-10 from MSN | Precision | 0.317 | 0.197 | 0.281 | 0.265 |
| | Recall | 0.426 | 0.404 | 0.361 | 0.397 |
| | F_1 | 0.364 | 0.265 | 0.316 | 0.315 |
| Top-10 from Google | Precision | 0.338 | 0.207 | 0.304 | 0.283 |
| | Recall | 0.457 | 0.43 | 0.392 | 0.426 |
| | F_1 | 0.389 | 0.279 | 0.342 | 0.337 |
| Top-10 from Yahoo | Precision | 0.33 | 0.198 | 0.31 | 0.279 |
| | Recall | 0.447 | 0.411 | 0.401 | 0.42 |
| | F_1 | 0.38 | 0.268 | 0.35 | 0.332 |
| Top-10 from MSN + Google + Yahoo | Precision | 0.246 | 0.385 | 0.348 | 0.326 |
| | Recall | 0.427 | 0.437 | 0.376 | 0.413 |
| | F_1 | 0.312 | 0.409 | 0.361 | 0.361 |
| Top-50 from MSN | Precision | 0.438 | 0.271 | 0.382 | 0.364 |
| | Recall | 0.366 | 0.348 | 0.305 | 0.34 |
| | F_1 | 0.399 | 0.305 | 0.339 | 0.348 |
| Top-50 from Google | Precision | 0.441 | 0.268 | 0.392 | 0.367 |
| | Recall | 0.395 | 0.368 | 0.335 | 0.366 |
| | F_1 | 0.417 | 0.31 | 0.361 | 0.363 |
| Top-50 from Yahoo | Precision | 0.435 | 0.275 | 0.4 | 0.37 |
| | Recall | 0.374 | 0.363 | 0.328 | 0.355 |
| | F_1 | 0.402 | 0.244 | 0.360 | 0.335 |
| Top-50 from MSN + Google + Yahoo | Precision | 0.377 | 0.575 | 0.483 | 0.478 |
| | Recall | 0.244 | 0.243 | 0.194 | 0.227 |
| | F_1 | 0.296 | 0.342 | 0.277 | 0.305 |

Table 6: Soft classification using top snippets using Rainbow on KDD Cup test set.

| | | Judge 1 | Judge 2 | Judge 3 | Overall |
|--|-----------|---------|---------|---------|--------------|
| Top-10 from MSN | Precision | 0.504 | 0.335 | 0.439 | 0.426 |
| | Recall | 0.136 | 0.139 | 0.113 | 0.129 |
| | F_1 | 0.214 | 0.196 | 0.18 | 0.197 |
| Top-10 from Google | Precision | 0.527 | 0.349 | 0.434 | 0.437 |
| | Recall | 0.143 | 0.145 | 0.112 | 0.133 |
| | F_1 | 0.225 | 0.206 | 0.178 | 0.203 |
| Top-10 from Yahoo | Precision | 0.534 | 0.357 | 0.47 | 0.454 |
| | Recall | 0.145 | 0.149 | 0.122 | 0.139 |
| | F_1 | 0.228 | 0.209 | 0.194 | 0.21 |
| Top-10 from MSN + Google + Yahoo | Precision | 0.385 | 0.574 | 0.484 | 0.481 |
| | Recall | 0.161 | 0.156 | 0.126 | 0.148 |
| | F_1 | 0.227 | 0.245 | 0.2 | 0.224 |
| Top-50 from MSN | Precision | 0.623 | 0.407 | 0.503 | 0.511 |
| | Recall | 0.168 | 0.169 | 0.13 | 0.156 |
| | F_1 | 0.265 | 0.24 | 0.207 | 0.237 |
| Top-50 from Google | Precision | 0.634 | 0.414 | 0.537 | 0.528 |
| | Recall | 0.172 | 0.172 | 0.139 | 0.161 |
| | F_1 | 0.271 | 0.242 | 0.221 | 0.244 |
| Top-50 from Yahoo | Precision | 0.609 | 0.416 | 0.538 | 0.521 |
| | Recall | 0.166 | 0.173 | 0.139 | 0.159 |
| | F_1 | 0.261 | 0.244 | 0.221 | 0.242 |
| Top-50 from MSN + Google + Yahoo | Precision | 0.449 | 0.657 | 0.547 | 0.551 |
| | Recall | 0.187 | 0.179 | 0.142 | 0.169 |
| | F_1 | 0.264 | 0.281 | 0.225 | 0.257 |

Table 7: Hard classification using top snippets using Rainbow on KDD Cup test set.

4.2.2 SVM classifier

With the motivation of trying to find a better classifier than Rainbow’s naive Bayes, we trained the SVM^{light} classifier [Joa98] with the same 111 queries and tested it on the same 800 queries. The set of features chosen were the same as that used for testing the 800 queries using the Rainbow classifier, namely top 50 and top 10 snippets from the three search engines.

Tables 8 and 9 show the evaluation results after testing the 800 queries using SVM^{light} . Performance obtained using the SVM classifier is noticeably worse than that obtained using the naive Bayes classifier.

| | | Judge 1 | Judge 2 | Judge 3 | Overall |
|--|-----------|---------|---------|---------|--------------|
| Top-10 from MSN | Precision | 0.271 | 0.126 | 0.249 | 0.215 |
| | Recall | 0.369 | 0.263 | 0.339 | 0.324 |
| | F_1 | 0.312 | 0.17 | 0.287 | 0.257 |
| Top-10 from Google | Precision | 0.279 | 0.134 | 0.256 | 0.223 |
| | Recall | 0.38 | 0.279 | 0.349 | 0.336 |
| | F_1 | 0.322 | 0.181 | 0.295 | 0.266 |
| Top-10 from Yahoo | Precision | 0.276 | 0.126 | 0.251 | 0.218 |
| | Recall | 0.376 | 0.264 | 0.327 | 0.322 |
| | F_1 | 0.318 | 0.171 | 0.284 | 0.258 |
| Top-50 from MSN | Precision | 0.29 | 0.141 | 0.261 | 0.231 |
| | Recall | 0.395 | 0.293 | 0.356 | 0.348 |
| | F_1 | 0.334 | 0.19 | 0.301 | 0.275 |
| Top-50 from Google | Precision | 0.297 | 0.147 | 0.272 | 0.239 |
| | Recall | 0.405 | 0.306 | 0.372 | 0.361 |
| | F_1 | 0.343 | 0.199 | 0.314 | 0.285 |
| Top-50 from Yahoo | Precision | 0.3 | 0.148 | 0.272 | 0.24 |
| | Recall | 0.409 | 0.309 | 0.354 | 0.357 |
| | F_1 | 0.346 | 0.2 | 0.308 | 0.285 |
| Top-50 from MSN + Google + Yahoo | Precision | 0.309 | 0.159 | 0.282 | 0.25 |
| | Recall | 0.646 | 0.331 | 0.59 | 0.522 |
| | F_1 | 0.418 | 0.215 | 0.382 | 0.338 |

Table 8: SVM-based soft classification using top snippets.

| | | Judge 1 | Judge 2 | Judge 3 | Overall |
|--|-----------|---------|---------|---------|--------------|
| Top-10 from MSN | Precision | 0.423 | 0.156 | 0.401 | 0.327 |
| | Recall | 0.115 | 0.065 | 0.109 | 0.096 |
| | F_1 | 0.181 | 0.092 | 0.171 | 0.148 |
| Top-10 from Google | Precision | 0.444 | 0.165 | 0.406 | 0.338 |
| | Recall | 0.121 | 0.069 | 0.111 | 0.1 |
| | F_1 | 0.19 | 0.097 | 0.174 | 0.154 |
| Top-10 from Yahoo | Precision | 0.428 | 0.141 | 0.403 | 0.324 |
| | Recall | 0.117 | 0.059 | 0.105 | 0.094 |
| | F_1 | 0.184 | 0.083 | 0.167 | 0.145 |
| Top-50 from MSN | Precision | 0.41 | 0.141 | 0.426 | 0.326 |
| | Recall | 0.112 | 0.06 | 0.116 | 0.096 |
| | F_1 | 0.176 | 0.084 | 0.182 | 0.147 |
| Top-50 from Google | Precision | 0.471 | 0.168 | 0.425 | 0.355 |
| | Recall | 0.128 | 0.07 | 0.116 | 0.105 |
| | F_1 | 0.201 | 0.099 | 0.182 | 0.161 |
| Top-50 from Yahoo | Precision | 0.469 | 0.169 | 0.435 | 0.358 |
| | Recall | 0.128 | 0.07 | 0.113 | 0.104 |
| | F_1 | 0.201 | 0.099 | 0.179 | 0.16 |
| Top-50 from MSN + Google + Yahoo | Precision | 0.5 | 0.169 | 0.433 | 0.367 |
| | Recall | 0.209 | 0.07 | 0.181 | 0.153 |
| | F_1 | 0.295 | 0.099 | 0.255 | 0.216 |

Table 9: SVM-based hard classification using top snippets.

5 Conclusion

From the experiments carried out using various features and their combinations, we have been successful in addressing the questions we posed at the start of the paper. We have shown that using simpler techniques we can achieve comparable performance to that achieved by the more complex techniques used in the KDD Cup 2005 competition. By comparing the performance obtained by using only the query terms versus using the enhanced features, we can say that enhancing the queries with related text does help in improving the query classification. By observing the results obtained, we can also conclude that the use of aggregated snippets to classify the queries provides better performance than using the page content and other combinations of features.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0328825. We also thank Xiaoguang Qi for his code and assistance.

References

- [BFG⁺07] Andrei Broder, Marcus Fontoura, Evgeniy Gabrilovich, Amruta Joshi, Vanja Josifovski, and Tong Zhang. Robust classification of rare queries using web knowledge. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 231–238, New York, NY, July 2007. ACM Press.
- [BJCF07] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, and Ophir Frieder. Varying approaches to topical web query classification. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 783–784, New York, NY, 2007. ACM Press.
- [BJF⁺05] Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David D. Lewis, Abdur Chowdhury, and Aleksander Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 42–49, Washington, DC, 2005. IEEE Computer Society.
- [Joa98] Thorsten Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.
- [KTB05] Zsolt T. Kardkovacs, Domonkos Tikk, and Zoltan Bansaghi. The Ferrety algorithm for the KDD Cup 2005 problem. *SIGKDD Explorations*, 7(2):111–116, 2005.
- [LZ05] Ying Li and Zijian Zheng. KDD Cup 2005. Online at <http://www.acm.org/sigs/sigkdd/kdd2005/kddcup.html>, 2005.
- [LZD05] Ying Li, Zijian Zheng, and Honghua (Kathy) Dai. KDD CUP-2005 Report: Facing a great challenge. *SIGKDD Explorations*, 7(2):91–99, 2005.
- [McC96] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. <http://www.cs.cmu.edu/~mccallum/bow>, 1996.

- [SPS⁺06] Dou Shen, Rong Pan, Jian-Tao Sun, Jeffrey Junfeng Pan, Kangheng Wu, Jie Yin, and Qiang Yang. Query enrichment for web-query classification. *ACM Transactions on Information Systems*, 24(3):320–352, July 2006.
- [SSYC06] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building bridges for web query classification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 131–138, New York, NY, 2006. ACM Press.