Analysis of Varying Approaches to Topical Web Query Classification

Steven M. Beitzel
Telcordia Technologies, Inc.
steve@research.telcordia.com

Eric C. Jensen, Abdur Chowdhury
Summize, Inc.
{ej,abdur}@summize.com

Ophir Frieder
Illinois Institute of Technology
ophir@ir.iit.edu

ABSTRACT

Topical classification of web queries has drawn recent interest from forums such as the 2005 KDD Cup because of the promise it offers in improving retrieval effectiveness and efficiency. Many proposed techniques make use of documents classified in taxonomies (such as the ODP: Open Directory Project – http://www.dmoz.org) to inform on the class of a web query. Implicit in these approaches is the assumption that topically classifying queries is equivalent to the general topical text classification task (although with few directly available features from such short queries). We test this assumption by comparing and combining classifiers trained directly from manually classified queries and their retrieved documents, trained from categorized documents in the ODP, and induced from unlabeled query logs for pre-retrieval classification. We find that training classifiers directly from manually classified queries outperforms the best general topical classifier by 48% in relative F1 score. We attribute this to a mismatch in task when applying a general classifier to queries. For example, a typically vague web query classified as “business” is likely to retrieve documents classified as “news” and “organizations” in addition to those labeled “business.” Equating a “business” class of queries with a “business” class of documents, then, is not appropriate.

1. INTRODUCTION

Topical web query classification is a key research problem in the arena of web studies. It is clear that effective, automatic methods for classifying web queries can be used by search services to improve their efficiency and effectiveness. Until recently, most automatic text classification research has been focused on classifying passages or documents [17]. These documents typically contain relatively large quantities of text that can be used to provide adequate training features for machine learning approaches [14]. The task of classifying web queries is fundamentally different, in that web queries are very short, providing machine learning algorithms with very few features for use in training [2].

The 2005 KDD Cup competition focused on the topical classification of web queries, with the goal of encouraging researchers to examine and develop approaches for this new task. Because of the lack of typical training data, many participants were forced to turn to external sources of information to provide training for their systems, including human-edited taxonomies of web pages, and training features extracted from the top retrieved documents for various queries [15]. Many participants performed well, achieving satisfactory scores for both classification precision and total performance (measured by F1, the harmonic mean of precision and recall), although little study was done on how well the external information used for classification actually fits the task of classifying web queries as opposed to documents. It is not clear if mapping the categorization of a query to existing categorizations of full-text documents is the most effective approach for topically classifying web queries.

In relevant previous work, Beitzel, et. al developed methods for automatic topical classification of web queries that do not require information from external sources [3-5]. Web search services in production often cannot afford to spare the temporal and computational resources that are required to harness external information in online taxonomies and retrieved documents. The disparity between these approaches and the approaches used in KDD cup seems to suggest that the task of automatic web query classification is not precisely defined. The optimal approaches are likely to vary, given the overall goals of the system (i.e., how are classifications going to be used), and operational requirements. Additionally, it is possible that the inherent concepts described by queries of a certain class (“news”, for example) may not fully overlap with concepts described by full-text documents deemed to be in that same class. Documents contain much more text, and are likely to be constrained to a countable number of defined subjects. Queries, on the other hand, are designed to retrieve documents about a topic of interest to the user, not necessarily to describe that topic fully. This disparity between the language of queries and the language of documents makes it difficult to know how to best apply knowledge from external resources to the task of classifying queries.

In this study, we set out to examine the conditions under which it makes sense to use various approaches to query classification. Specifically, we study the following research questions:

- How do the distinguishing features in real web queries compare to those found in retrieved documents associated with those queries?
- When does it make sense to map an online taxonomy to the set of categories used for classification?
- Which learning algorithms or classification approaches are likely to perform best under varying circumstances?
- What is the most optimal method for combining separate classification approaches?

In section 2 we give an overview of related work on automatic topical query classification. In section 3 we describe the details of our experimental methodology. In section 4 we present our experimental results, and give an analysis of our findings.
Finally, in section 5 we state our conclusions and give directions for future work in this area.

2. PRIOR WORK

Until recently, the vast majority of work in automatic text classification has been focused on passages or full documents. For these cases, where there is an ample supply of available features, there have been several studies proposing well-defined learning approaches. Sebastiani performed a recent survey of such techniques in [17]. As the focus has shifted to automatic classification of web queries, researchers have had to go beyond traditional techniques, owing to the fact that web queries are typically much shorter than documents and passages (between two and three terms, on average). Also, queries are often the user’s intent to distill their information need down into a just a couple of terms. To successfully classify queries, then, we must be able to capture the full scope of the user’s information need using some means other than simply the terms in the query itself. Some previous research has focused on clustering related queries [1] and classifying queries into non-topical categories, such as the user’s target task [6, 12] and geographical location [9], but for this study we focus our scope on topical classification only.

The ACM Conference on Knowledge and Data Discovery (KDD) holds an annual competition known as the “KDD Cup”. The task varies from year to year, usually focusing on an area in the information sciences. In 2005, the task was topical query categorization. The dataset consisted of 800,000 web queries, and 67 possible categories, with each category pertaining to a specific topic (“Sports-Baseball” is one example of a KDD cup category). Each participant was to classify all queries into as many as five categories. An evaluation set was created by having three human assessors independently judge 800 queries that were randomly selected from the sample of 800,000. On average, the assessors assigned each query to 3.2 categories for their judgments. Once complete, these evaluations were used to calculate the classification precision and F1 score for each submission. Participants were not restricted in regards to the use of particular resources to aid in making classification decisions. As a result, several runs made use of various forms of external information.

Shen and colleagues used an ensemble approach of several different classification techniques to create the winning submission for the 2005 KDD Cup [18, 19]. They built synonym-based keyword-matching classifiers to map the category hierarchies used by search engines to the one employed at the KDD Cup. They extend the keyword matching to include various grammatical forms, and also via WordNet (http://wordnet.princeton.edu). Three synonym-based classifiers were built, using the category hierarchies from Google™, Looksmart™, and an internal search engine based on Lemur [8], searching a crawl of the ODP hierarchy. They also built a statistical classifier using SVM_light [11]. They collected training data by using the mappings found for the synonym classifiers to find pages in the relevant ODP categories for each query. The terms from the pages, their snippets (query-biased summaries), titles, and their category names were stemmed and processed for the removal of stopwords and used to train the SVM classifier. They also used two ensemble classifiers, one using the 111-query training set to learn weights for each component classifier, and one giving equal weight to each component classifier. This approach resulted in an F1 score that outperformed all other participants by nearly 10%.

Kardkovacs, et. al proposed an approach called the Ferrety Algorithm [13]. Their approach was similar to that of the winning team, employing the Looksmart and Zeal search engines to build their taxonomy mapping, enriching the mapping process with stemming, stopword removal, and the application of WordNet. They used this data to train their own learning software and make classification decisions, and they experimented with several different methods of feature selection, achieving good results on the KDD cup data.

3. METHODOLOGY

On the task of topical query classification, we compare and combine query classifiers that can be applied before gathering the retrieved documents, a general text classifier trained with documents in a directory, and direct query classifiers trained on the retrieved documents of classified queries. Beitzel and colleagues in previous work built classification methods from query logs that could be applied before retrieving documents. We include their exact methods in our comparison. The top performers in KDD Cup 2005 all used approaches that were variations on a general theme: train a general text classifier from an external taxonomy of web pages and map that taxonomy’s categories to those of the queries. As a baseline, we apply such an approach by training a support vector machine with the ODP and manually mapping its categories to our queries’ categories. Terms from retrieved documents are used as the feature set during classification. Finally, we examine the difference between these methods and that of an SVM built directly with the same categories for which it is tested; trained using the queries’ categories as their classes and terms from their retrieved documents as the features.

We use the 20,000 queries manually classified into 18 general topical categories available detailed in previous work by Beitzel, et al. We partitioned the dataset into 1/3 training, 1/6 tuning, and 1/2 testing. For the SVM classifiers, training data was used to build the model and tuning data to select the threshold at which we report F1 on testing. Although the generic classifier is not trained using the labels from our training set, we use the retrieved documents from the training set (without reference to any particular query) and their labels in the ODP to build the model. The pre-retrieval methods are not trained from our data, but rather only require a tuning set to select the optimal threshold. Therefore, these simply use the entirety of the training and tuning sets combined to set this threshold.

For our methods which use them, we processed each of the 20,000 queries with Google to obtain the top ten retrieved documents and their snippets. We then crawled the full text of each of these retrieved documents, parsed the HTML to extract the text, and performed only very basic case normalization before counting the frequency of each unique term.

Our generic text classifier and classifier learned directly from retrieved documents each use the same configuration of the libsvm package [7]. We use linear kernels, default parameters,
and the voting one-vs-one method of addressing the multiclass problem: building a binary classifier for each pair of classes. Feature values are linearly scaled between 0 and 1 using the svm-scale program from libsvm, and for classifiers using the text of retrieved documents F-score feature selection is performed using their fselct script to reduce the number of features to roughly 1/3 of the original 4.8 million [10].

3.1 Pre-retrieval Classifiers from User Logs
To evaluate and compare the performance of topical query classifiers that make use of external resources (such as human-edited taxonomies or features extracted from the retrieved documents) to the performance of classifiers that train on queries alone, we employed classification approaches developed in recent previous work by Beitzel and colleagues in our experiments. They proposed three classifiers, each using distinctly different techniques, to classify a large portion of the query stream. Their baseline approach involved doing an exact-match lookup of an unseen query into a large database of queries that had been previously classified into 18 general topical categories by a group of human assessors. They found, not surprisingly, that such an approach yielded relatively high precision, but very low recall. To mitigate this, they trained a perceptron learner on the large database of classified queries in an attempt to learn to distinguish between categories, giving slightly improved performance. As a final technique, they used selectional preferences [16] to extract classification rules from a large unlabeled log of web queries. This gave a large increase in classification recall, significantly improving overall performance. To evaluate all of the classification approaches used in this study, as well as compare to previous work, we also use the test collection developed by Beitzel and colleagues. Specifically, they created a test set of 20,000 web queries, manually classified by human assessors into the same 18 topical categories used in the older, larger collection described above.

3.2 Generic Text Classifier from ODP
The first post-retrieval classifier in our comparison makes the assumption made implicitly by most in KDD Cup 2005: That we can topically classify queries by treating their terms and the terms of their retrieved documents as a text as in any other topical text classification task. With this assumption in place, the model can be induced from any topically classified training texts. We built such a model using the ODP categories, manually mapped to those of our query classification task, as labels and the full text of documents in those categories as features. Although these documents were spread across thousands of very specific ODP categories, for most one of the general parent categories in which they reside seemed to correspond reasonably to one of our 18.

To ensure each classifier in our comparison is provided with comparable amounts of training data we used only those documents retrieved by our set of training queries, looked up their categories in the ODP, manually mapped each of those categories to one of ours, and built the model using each document as an instance without regard to query. Of course, not all retrieved documents are present in the ODP, and some are present in multiple categories. As in our other methods, when an example has multiple classes we treat it as multiple training instances, one for each class. This resulted in 7,549 training instances, not far from the 1/3 of our collection used elsewhere.

3.3 Classifiers Learned Directly from Retrieved Documents
Finally, we examine classifiers built in the conventional manner: training directly on a subset of the same dataset they are tested on. Here, we explicitly learn models from the manually classified queries, using each query as a training instance. The only uncommon challenge in this technique is the lack of features inherent in a query which is 2-3 terms on average. To overcome this, we use the snippets and text of the documents retrieved by each query to expand its feature set. Unlike with the generic classifier where we always used the full text of the documents because that corresponded with the way the model was built, here we also experiment with a classifier learned from and classified with only the snippets.

4. RESULTS & ANALYSIS
To determine how our three categories of query classifiers compare to each other, we perform three types of analysis. First, we examine the overall optimal performance for each classifier. Next, we combine the classifiers to try and exploit their differences for overall improved performance. Finally, we examine these differences in further detail by detailing the specific errors made by each classifier.

4.1 Individual Classifier Performance
The performance of each classifier over our 10,000 query testing set using the threshold of optimal F1 from the tuning set is detailed in Table 1. Surprisingly, one can achieve roughly equivalent performance from pre-retrieval classifiers that use only the query string itself for classification as that of the generic text classifier which requires the retrieved documents. The post-retrieval classifiers learned directly from classified query logs improve upon this substantially, with a 48% relative improvement in F1. Clearly, performance is lost when treating query classification as the general topical text classification problem by mapping document taxonomies to query ones, and interchanging documents as training instances with sets of retrieved documents as testing ones.

<table>
<thead>
<tr>
<th>Pre-retrieval query classifiers</th>
<th>Micro. Precision</th>
<th>Micro. Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact match</td>
<td>0.2959</td>
<td>0.0991</td>
<td>0.1484</td>
</tr>
<tr>
<td>perceptron</td>
<td>0.2030</td>
<td>0.2777</td>
<td>0.2346</td>
</tr>
<tr>
<td>SP</td>
<td>0.1698</td>
<td>0.3671</td>
<td>0.2322</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generic text classifier</th>
<th>Micro. Precision</th>
<th>Micro. Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP</td>
<td>0.2750</td>
<td>0.2580</td>
<td>0.2662</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Direct query classifiers</th>
<th>Micro. Precision</th>
<th>Micro. Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>snippets</td>
<td>0.3364</td>
<td>0.4757</td>
<td>0.3941</td>
</tr>
<tr>
<td>snippets and docs</td>
<td>0.3947</td>
<td>0.3704</td>
<td>0.3822</td>
</tr>
</tbody>
</table>

Table 1: Individual Classifier Performance
4.2 Combining Classifiers

Based on the results from the individual classifiers, we chose preference-ordered fusion to combine their results [3]. This method simply uses the classifications from higher-precision classifiers if they offered one with confidence above a threshold, but backs off to higher-recall classifiers when they did not. Since exact match and perceptron are high precision, but low recall, they were placed first in the preference order. The results of this combination are in Table 2. Despite their very different focus, combining pre-retrieval classifiers with the post-retrieval snippets one does not provide substantial improvement. With the additional information available post-retrieval, the imprecision of the pre-retrieval techniques prevents them from adding value.

<table>
<thead>
<tr>
<th></th>
<th>Micro. Precision</th>
<th>Micro. Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact match + perceptron + SP</td>
<td>0.1908</td>
<td>0.3216</td>
<td>0.2395</td>
</tr>
<tr>
<td>exact match + perceptron + SP + snippets</td>
<td>0.3416</td>
<td>0.4718</td>
<td>0.3963</td>
</tr>
</tbody>
</table>

Table 2: Combined Classifier Performance

However, these combinations do represent the best pre and post-retrieval performance we achieve. To examine this in more detail, Figure 1 includes the overall precision/recall tradeoffs. The ability of retrieved document classifiers to achieve greater recall than the log-based pre-retrieval ones is to be expected. However, the higher precision they’re capable of yielding at low recall is a result of their being trained and tested on the same nature of their errors show s many other classes being misclassified as them, suggesting tuning the thresholds on a per-category basis could improve performance. Conversely, holidays is very rarely classified into at this threshold. More interestingly, the overlap between categories becomes strikingly apparent: across classifiers holidays and travel are often confused, home and shopping, etc.

4.3 Failure Analysis

Finally, we focus on the particular errors made by each of our three types of classifiers. We examine the confusion matrices; with counts of classifications where rows are the true classes and columns are the predicted class for each test example (so the diagonal are correct classifications while the remaining are errors). For readability, we highlight rates above 40% in dark gray, 20% in medium, and 5% in light. Figure 2 shows errors for the best combined pre-retrieval classifier. Errors made by the generic ODP-based classifier seems to have a different idea of the topic “home,” for example, while the pre-retrieval classifier has one of “org”. The various classifiers strengths at different categories continue to suggest that some sort of fusion may be warranted.

Across all three figures, we see that some classes are better defined than others. Places, shopping, other, and business cause difficulty for all three classifiers. Specifically, the columnar nature of their errors shows many other classes being missclassified as them, suggesting tuning the thresholds on a per-category basis could improve performance. Conversely, holidays is very rarely classified into at this threshold. More interestingly, the overlap between categories becomes strikingly apparent: across classifiers holidays and travel are often confused, home and shopping, etc.

Figure 1: Best Pre- and Post-Retrieval Precision and Recall

Examining across matrices, the shortcomings of mapping the learned taxonomy onto the target one for categorization becomes apparent. The generic ODP-based classifier seems to have a different idea of the topic “home,” for example, while the pre-retrieval classifier has one of “org”. The various classifiers strengths at different categories continue to suggest that some sort of fusion may be warranted.

Figure 2: Combined exact match + perceptron + SP Classifier Confusion Matrix

Figure 3: ODP Classifier Confusion Matrix
5. CONCLUSIONS & FUTURE WORK

We have evaluated and analyzed three differing approaches to topical web query classification over a large, manually classified test collection. Our experiments suggest a mismatch in task between classifying documents and classifying web queries. We have found that for query classification, training directly with features from the queries themselves and from the documents those queries retrieve outperforms other approaches that use external resources for classification, such as those used in KDD Cup 2005 by as much as 48% in F1. Although our preference-ordered fusion of multiple approaches did not yield improved performance, further analysis does show substantial difference between the three methods. In future work, we will experiment with tuning thresholds on a per-category basis and developing improved combination strategies.

6. REFERENCES


