Query Classification Using Asymmetric Learning

Zheng Zhu
School of Computer Science and Information Systems
Birkbeck College, University of London
zheng@dcs.bbk.ac.uk

Mark Levene
School of Computer Science and Information Systems
Birkbeck College, University of London
mark@dcs.bbk.ac.uk

Ingemar J Cox
Department of Computer Science
University College London
ingemar@ieee.org

Abstract

Understanding the meaning of queries is a key task which is at the heart of web search. Classification of users' queries is a challenging task due to the fact that queries are usually short and often ambiguous. A common approach to tackle the problem of short and noisy queries is to enrich the queries. Various enrichment strategies have been proposed that are based on either pseudo-relevance feedback or secondary sources of information. In general, pseudo-relevance feedback based algorithms exhibit superior performance. However, in this case query classification can only occur after performing the retrieval, as the result set is needed to apply pseudo-relevance feedback.

Since some applications may prefer to perform query classification prior to, or in parallel with retrieval, there is a need to improve the performance of query classification based on secondary sources. In this paper, we present a hybrid strategy, in which training is based on pseudo-relevance feedback, but testing is based on a secondary source, specifically Yahoo's "suggested keywords". These keywords are based on co-occurrence data across queries. The classifier, which is built offline with training data, makes use of the top-n results during training, but not duing testing. Thus, there is an asymmetry between the training and testing data. We compared the classification using symmetrical and asymmetrical approaches on a large AOL search log. Symmetric training and testing using queries enriched with Yahoo keywords yielded a microaveraged F1 score of 44%. Asymmetric training (enriching with the top-10 Google snippets) and testing (enriching with Yahoo suggested keywords) increased the F1 score to 46%. This is comparable with a symmetric approach based on feedback of the top-2 pseudo-relevant documents, in which a similar number of enrichment terms is added.

1 Introduction

There are various applications that motivate query classification. These include paid placement advertising, classification/clustering of query results, query expansion, personalized search, and federated search. Query classification is a difficult task because queries usually consists of only a few terms, often leading to significant ambiguity.

To alleviate the problem of short queries, it is common to "enrich" the query with supplementary keywords, during both the training and testing phases. The supplementary keywords are derived from one of two broad sources. The first source is a form of pseudo-relevance feedback in which it is assumed that the top-n documents retrieved in response to the query are relevant. Typically, the snippets associated with the top-n documents, and provided as part of the retrieved results, are used as the source of enrichment. This is discussed in more detail in Section 3. The second source of enrichment data comes from related information such as a thesaurus, or co-occurrence information present in search engine query logs.

Previous work [2] and our own experiments indicate that better performance is achieved based on pseudo-relevance feedback. However, in some applications it may be desirable to perform query classification prior to, or in parallel with retrieval. For example, for paid placement advertising, it may be useful to perform query classification and the associated auction in parallel with the search. And for federated search, if query classification is used to select which databases to access, then classification cannot be based on retrieval results. The motivation for our work is to study the performance of query classification in the absence of pseudo-relevance feedback.

In this paper we investigate the performance of a query classification algorithm that uses a secondary source of information, namely Yahoo's "suggested keywords", for query enrichment during testing. The main novelty of this work is the use of pseudo-relevance feedback data during the training phase. That is, during training, we use retrieved Google snippets to enrich the query (pseudo-relevance feedback), but during testing/deployment, we use Yahoo's suggested keywords for query enrichment. Section 3 describes the procedure in detail. The experimental results, see Section 4.3, indicate an improvement in performance over a symmetric learning strategy. Specifically, if we train and test using Yahoo's suggested keywords, (symmetric training and testing), we achieve a classification rate of 44%. In contrast, asymmetric training and testing achieves a classification rate of 46%.

The paper is organised as follows: In Section 2 we describe related work on query classification, and in Section 3 we elucidate the methodology of our approach. In Section 4 we describe the data set we used in this experiment, the experimental procedure and the results we obtained, and we conclude with some remarks in Section 5.

2 Related Work

There are three broad research themes within the query classification community. The first theme relates to the acquisition of training data. In many cases, labelled training data is sparse and/or expensive to acquire. To alleviate this, researchers have investigated using alternative sources of training data [6, 4]. This kind of alternative training data may not be optimal, for example, in the absence of a direct mapping between the class structures. For example, some researchers have included training data from the Open Directory Project or the Yahoo! Web Directory, which has been used in web page classification tasks.

The second theme relates to various different ways to enrich queries. There have been many proposals, some of which use pseudo-relevance feedback [6, 4], while others use secondary information sources [5]. Empirical results indicate that query enrichment can significantly improve performance [2]. And query enrichment based on pseudo-relevance feedback is usually superior to enrichment based on secondary sources. However, despite this superiority, there is a need to perform query classification prior to performing the retrieval, which precludes the use of enrichment based on pseudo-relevance feedback.

The third theme relates to the number of classes in the query classification taxonomy. In many cases, the goal has been to classify a query into one or more of several broad classes. In this case the number of classes is typically less than 100. For example, Beitzel *et al.* [3] classify test queries into 18 classes , while for the 2005 KDDCUP, there were 67 predefined categories [6]. In contrast, other work, motivated by paid placement advertising, requires a much larger number of classes. For example, the taxonomy used in [4] has 6000 classes.

Our work focuses on the second theme, i.e., query enrichment. Our data set is derived from the same source, i.e. AOL, as that used in Beitzel *et al* [3]. Their method made use of a computational linguistics approach, called selectional preferences, combined with perceptron training and exact matching, which they call the *pre-retrieval* approach, since it does not require pseudo-relevance feedback. In a later refinement [2], they found that enriching the queries with snippets from search engine results outperformed their pre-retrieval solution, increasing the F1 score by 15%. As the experimental conditions of [2] are similar to ours, we compare our results to theirs in Section 4.3.7.

3 Methodology

In this section, we describe two strategies for query enrichment. The first is based on pseudo-relevance feedback, see (Section 3.1), using Google snippets. The second is query enrichment based on Yahoo's suggested keywords, see Section 3.2.

Section 3.3 then describes the classifier used in our experiments. Section 3.3.1 then describes the data representation used and Section 3.3.2 describes details relating to the use of support vector machines for multi-label multi-class classification. Section 3.4 describes our evaluation criteria, based on microaveraged precision, recall and F1.

3.1 Pseudo Relevance Feedback

Pseudo-relevance feedback is a commonly employed approach for enrichment. It assumes that the top-*n* documents in the retrieved result set are relevant to a user's information needs; here the documents refer to snippets or full web pages of retrieved URLs and are represented within the vector space model (Section 3.3.1). Here we assume that the retrieved documents are snippets returned as part of the search engine result set. Note that the work of [6] uses a similar methodology.

3.2 Term Co-occurrence from Query Logs

User query logs provide a valuable source for query classification. For example, the query "machine learning" is strongly correlated to "machine learning algorithm" and "machine learning research" found in query logs. The correlated terms provide an alternative method for query enrichment. Rather than compute the co-occurrence information directly from a query log, we used Yahoo! Related Suggestions[1], which are based on query logs, as the secondary source of enrichment. In practice, term co-occurrence information can be performed prior to search, while pseudo-relevance feedback can only be performed after performing the search.

3.3 Classifier

Support vector machines (SVM) are a classification technique that has been successfully applied to many text classification problems. In this work we apply SVMs to query classification, and make use of the Liblinear[2] toolkit, which is an open-source package for solving large-scale regularized linear classification problems.

3.3.1 Data representation

We represent a document by the vector

$$\overrightarrow{d_i} = (log(1+f(w_1, d_i))*idf_1, \ldots, log(1+f(w_n, d_i))*idf_n), \quad (1)$$

where $f(w_j, d_i)$ is the frequency of term w_j in document d_i, and idf_k is defined by

$$idf_k = log(\frac{|D|}{df_k}),$$

where df_k is the number of documents in the collection which contain the term w_k and $|D|$ is the cardinality of the document collection.

In the basic model, known as a uni-gram model, each feature represents a single term and is independent of other features. We can extend the uni-gram model by using pairs of adjacent terms as additional features, to obtain a model known as a bi-gram model. This leads to the *bi-gram SVM* (BSVM), which we use here. (In our experiments a bi-gram model includes uni-gram data.)

Each user query gives rise to a collection of snippets, which leads to two ways of representing an enhanced query vector, i.e.

$$\overrightarrow{q} = \frac{\sum_{i=1}^{N} \overrightarrow{d_i}}{N} \quad (2)$$

and

$$\overrightarrow{q} = \sum_{i=1}^{N} \overrightarrow{d_i}, \quad (3)$$

where N is the number of snippets, and $\overrightarrow{d_i}$ is the document vector for the ith snippet. Equation 2, which averages the snippet vectors, is denoted by *AS*, while Equation 3, which sums the snippet vectors, is denoted by *SS*. The enhanced query vector $\overrightarrow{q}$ conveys the information of original query in vector space model and is used as the input for our classifier.

3.3.2 Multi-label, Multi-class Classification

Query classification is inherently a multi-class, multi-label problem. However, an SVM is inherently a two-class classifier. To reduce a multi-class problem to a two-class problem, a common technique is to build multiple one-versus-the-rest classifiers with one category being the positive class and the remaining categories being the negative classes. This method will cause an *unbalanced* class distribution as the negative class is usually much larger than the positive class. In order to make it *balanced*, it is usual to set several penalty parameters in the SVM formulation. The SVM toolkit, Liblinear, implements such a penalty function, which we use by setting a parameter, *w*, with a value equal to the ratio of each class size.

Query classification is also a multi-label problem [6], since each query can potentially be associated with several labels. When we convert a multi-class problem into a two-class problem, the multi-label problem can be solved simultaneously. If a query is associated with $|c|$ number of labels, it is treated as a positive query when we train or test a model for one of the $|c|$ classes.

3.4 Evaluation Criteria

To evaluate the performance, we use microaveraged precision, microaveraged recall and microaveraged F1. In the two-class case with *positive* and *negative* classes, the *true positives* (TP) and *true negatives* (TN) are data correctly classified as *positive* and *negative* respectively. A *false positive* (FP) occurs when the outcome is incorrectly predicted as *positive* when it is actually *negative*. A *false negative* (FN) occurs when the outcome is incorrectly predicted as *negative* when it is actually *positive*.

For the multi-class problem, the microaveraged performance measures are given by

$$Microprecision = \frac{\sum_{i=1}^{|C|} |TP|_i}{\sum_{i=1}^{|C|} |TP|_i + \sum_{i=1}^{|C|} |FP|_i}, \quad (4)$$

$$Microrecall = \frac{\sum_{i=1}^{|C|} |TP|_i}{\sum_{i=1}^{|C|} |TP|_i + \sum_{i=1}^{|C|} |FN|_i}, \quad (5)$$

$$MicroF1 = \frac{2 \times Microprecision \times Microrecall}{Microprecision + Microrecall}, \quad (6)$$

where $|C|$ is the number of the classes, $|TP_i|$ is the number of *true positive* for positive class *i*, $|FP_i|$ is the number of *false positive* for positive class *i* and $|FN_i|$ is the number of *false negative* for positive class *i*.

Although we report recall, precision and F1, our discussion focuses on the F1 measure.

[1] http://developer.yahoo.com/search/web/V1/relatedSuggestion.html

[2] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

4 Experiments Description

In Subsection 4.1 we describe the data set used, while in Subsection 4.2 we explain the experimental setting. Finally, in Subsection 4.3 we describe the experimental results.

4.1 Experiment Data Set

We made use of two manually classified subsets of an AOL search log [3]. The first one contains 9,913 manually classified queries resulting from a Master's level Information Science class assignment at Bar-Ilan University during 2007. The participants had some knowledge of classification, cataloguing and indexing. The ontology we have adopted for classification consists of 31 top level categories, which constitutes a reasonable searcher's ontology, The 31 categories are listed in the "Category" column of Table 1. However in our experiments, we discarded four classes(*url*, *misspelling*, *noise*, and *other*) which do not contain topic information at the semantic level; for more detail, we refer the reader to [1]. We also incorporated labeled log data from AOL's research lab, which uses a similar ontology to ours. Table 1 tabulates that mapping of the AOL ontology to the one we use. After we remove duplicate queries, 17,862 distinct queries remain, and these form our labeled data set.

4.2 Experimental Setting

Table 2. Query enrichment terminology

Notation	Description
q	query term without any enrichment
s	Enriched query with Related Suggestions from Yahoo!
(*n*)g	Enriched query with top *n* Google results, here n is 2, 5, 10 or 20, respectively, i.e., 2g, 5g, 10g, 20g

The notation we employ for the experiments is shown in the Table 2. We used the settings in Table 2 to enrich both the training and test data. Unless otherwise stated, the data was represented as averaged snippets (Equation 2), using a balanced class setting and a bi-gram SVM. The evaluation is based on standard 10 fold cross-validation.

4.3 Results

In the following tables TE and TR denote testing and training data, respectively, and the performance measures are microaveraged.

4.3.1 Comparison of difference data representation

The first question we consider is which data representation is better. Averaged snippets (AS, see Equation 2) or summed snippets (SS, see Equation 3). Table 3 shows that AS significantly[3] outperforms SS.

Table 3. The results for different data representations

Data rep.	TE	TR	recall	precision	F1
AS	10g	10g	52.57%	59.70%	**55.91%**
SS	10g	10g	44.97%	55.76%	49.79%

The discrepancy between the two representation is due to the nonlinear transformation in Equation 1.

4.3.2 Bi-gram SVM versus Uni-gram SVM

In Section 3.3.1 we mentioned that a bi-gram model has more features than a uni-gram model, and that the former has generally achieved better performance on text classification problems. To verify this, we evaluated the two classifiers using 10g data (i.e. the top-10 snippets returned by Google in response to the query). The results are shown in Table 4.

Table 4. The results for uni-gram and bi-gram SVM

Model	TE	TR	recall	precision	F1
uni-gram	10g	10g	53.76%	50.64%	52.16%
bi-gram	10g	10g	52.57%	59.70%	**55.91%**

The experimental results confirm that the bi-gram SVM outperforms the uni-gram SVM, albeit at the expense of a larger feature space.

4.3.3 Balanced Class versus Unbalanced Class

The class size distribution is important for many classifiers. For example, if one class size is very small, then the classifier will assign all data to the large class in order to minimize the error rate,. To alleviate this problem, the penalty of misclassifying the data to the large class is increased. We refer to this as balancing. A comparison of balanced versus unbalanced classification is given in Table 5. As expected, balancing the class distribution leads to significantly better results. Note, however, that the precision of the balanced class is worse than the unbalanced class.

[3] This was tested by a paired-sample t-test at the 5% significance level.

Table 1. The set of classes and the mapping between our classes and those of AOL.

	Category	AOL Category		Category	AOL Category
1	Art		17	Misspelling	Misspellings
2	Auto	Autos	18	Nature	
3	Companies/Business	Business	19	News	News
4	Computing	Computing	20	Noise	
5	Directories		21	Other	Other
6	Education		22	People	
7	Employment		23	Places	Places
8	Entertainment	Entertainment	24	Pornography	Porn
9	Finance&Economy	Personal Finance	25	Religion	
10	Food and drink		26	Science	Research
11	Games	Games	27	Shopping	Shopping
12	Government, organizations and non-profit institutions	organization and institutions	28	Society& Community	
13	Health and Medicine	Health	29	Sports	Sports
14	Holiday	Holidays	30	Technology	
15	Home	Home	31	URL	URL
16	Law & Legislation		32		Travel

Table 5. The results of balanced and unbalanced classes

Class dist.	TE	TR	recall	precision	F1
unbalanced	10g	10g	42.26%	71.53%	53.13%
balanced	10g	10g	52.57%	59.70%	**55.91%**

4.3.4 The Random Classifier

To assist in assessing the performance of our system, consider the performance of a random classifier, which randomly assigns a label to each user query. Given that we have 27 classes, the probability of a *true positive*, *false positive* or *false negative* is $\frac{1}{27}$, $\frac{26}{27}$ and $\frac{26}{27}$, respectively. So for a random classifier, precision and recall are 3.7%.

4.3.5 Symmetrical learning using relevance feedback for query enrichment

We first consider the case in which we enrich the query using relevance feedback during both training and testing, i.e. symmetrical learning. In this case we averaged the term weights from the top-*n* snippets to construct an enhanced query vector, as described in Section 3.3.1, for both the training and test data.

The experimental result for this baseline is shown at Table 6, for various numbers of pseudo-relevant documents ranging from 2 to 20.

Table 6. Classification results using symmetric training and testing with enrichment based on various numbers of pseudo-relevant snippets retrieved from Google.

TE	TR	recall	precision	F1
2g	2g	47.22%	47.46%	47.32%
5g	5g	51.42%	57.06%	54.09%
5g	5g(+live data)	52.96%	56.45%	54.65%
10g	10g	52.57%	59.70%	55.91%
10g	10g(+live data)	54.06%	59.13%	**56.47%**
20g	20g	51.65%	60.62%	55.78%

The results in Table 6 show that best performance is obtained when the query is enriched using the top-10 snippets from the Google retrieved set. For 10g/10g, the F1 measure is approximately 56%. Note for training enrichment with 5g and live data, we only augment the original training data with manually classified live query data. But the improvement is not significant here.

4.3.6 Symmetrical Learning using Yahoo's suggested keywords

Here we consider the performance when we enrich the query using a secondary source of information, namely Yahoo's suggested keywords, row *s*/*s*[4] in Table 7. For comparison, Table 7 also includes the resuls when no enrichment is

[4] This results is for queries which have related suggestions

applied (row *q*/*q*), and the case when only the top-2 results are used for pseuo-relevance feedback (row 2g/2g). The latter case is included because the number of enrichment terms is similar to that provided by Yahoo suggested keywords. Significantly more terms are available if 5, 10 or 20 documents are used for enrichment using pseudo-relevance feedback.

Table 7. Classification results using (i) no enrichment, (ii) enrichment based on Yahoo's suggested keywords, and (iii) the top-2 Google snippets. Training and testing is symmetric.

TE	TR	recall	precision	F1
q	q	53.78%	17.01%	25.83%
s	s	50.00%	39.14%	43.89%
2g	2g	47.22%	47.46%	**47.32%**

As expected, Table 7 shows performance without enrichment (*q*/*q*) is worse. Enrichment using Yahoo's suggested keywords significantly improves performance, from an F1 score of about 26% to almost 44%. However, enriching with only the top-2 pseudo-relevant documents is superior (about 47%). And enrichment using 10g/10g (Table 6) is much better, with an F1 score of almost 56%.

We now investigate whether asymmetrical learning can improve performance.

4.3.7 Asymmetrical Learning

In this approach we enrich the query using pseudo-relevance feedback of the top-10 Google snippets during training, but only enrich the query with Yahoo's suggested keywords during testing.

Table 8. Asymmetric training and testing. Training is performed using queries enriched with either (i) the top-10 Google snippets (10g), or (ii) the top-10 Google snippets and Yahoo suggested keywords (10g+s). Note that row 10g/10g represents symmetric training and testing and differs from the score reported in Table 6 due to pruning of the test set (see text for details.)

TE	TR	recall	precision	F1
s	10g	37.84%	57.54%	45.65%
2g	10g	41.58%	59.10%	48.81%
10g	10g	52.74%	58.42%	**55.43%**
s	10g+s	38.28%	57.10%	45.83%

Table 8 indicates that performance using Yahoo's suggested keywords during testing is improved from an F1 score of 44% to an F1 score of 46%. This improvement is statistical significant at the 5% significance level. Note that Yahoo's suggested keywords does not provide suggestions for all queries in our test set. For our experiments, training was performed with *all* queries, as before. However, testing was only conducted on queries that could be enriched. This reduced the number of queries from 17,862 to 7,654. For comparison purposes, Table 7 also include a row (10g/10g) reporting the scores for classification on this reduced test set.

Comparing our results to Beitzel *et al.* [2] we note the following. Beitzel *et al.* report an F1 measure of 24% when no enrichment is applied. This is consistent with our experimental results, reported in Table 7. When queries were enriched using the top-10 Google snippets they reported an F1 score of 39.6%, which is much less than the performance of our classifier, i.e. (10g, 10g). This is probably due to the fact that Beitzel *et al.* perform hold-out evaluation with 1/3 data for training while we use 10-fold cross validation with 9/10 data for training.

5 Concluding Remarks

In some applications, e.g. federated search, there is a need to perform query classification prior to performing the search. In such a case, enriching the query with results derived from pseudo-relevance feedback is not possible. Without pseudo-relevance feedback, query enrichment must be accomplished using secondary sources of information.

In this paper, we considered the use of Yahoo's suggested keywords as such source of secondary information. As expected, the performance, as measured by microaveraged F1, is significantly less than can be obtained using pseudo-relevance feedback. Specifically, our experiments showed that using a symmetric testing and training procedure, enriching with Yahoo's suggested keywords produced an F1 score of about 44%, which is substantially less than enriching with the top-10 snippets from Google, where the F1 score is about 56%. Even if only enriching with a similar number of terms, i.e. using the top-2 snippets, the F1 score is 47%, which is about 3% better than using Yahoo suggested keywords.

To improve performance, we investigated an asymmetric learning strategy. Training was performed with queries enriched using the top-10 snippets from Google, but testing was performed using queries enriched using Yahoo's suggested keywords. In this case, performance increased from 44% to 46%. This is well below that which can be obtained using a symmetric learning procedure and enrichment based on the top-10 snippets. However, if we compare to the performance achieved when only the top-2 snippets are used for enrichment, i.e. a similar number of enrichment terms are used, the performance is very comparable. This suggests that if a sufficiently rich source of secondary information is available, it may be possible for enrichment based on secondary sources to be competitive with those based on pseudo-relevance feedback. This is a topic of future work.

References

[1] J. Bar-Ilan, Z. Zhu, and M. Levene. Topic-specific analysis of search queries. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pages 35–42, New York, NY, USA, 2009. ACM.

[2] S. M. Beitzel, E. C. Jensen, A. Chowdhury, and O. Frieder. Varying approaches to topical web query classification. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 783–784, New York, NY, USA, 2007. ACM.

[3] S. M. Beitzel, E. C. Jensen, O. Frieder, D. D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 42–49, Washington, DC, USA, 2005. IEEE Computer Society.

[4] A. Z. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. 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, USA, 2007. ACM.

[5] H. Jian, W. Gang, L. Fred, S. Jian-tao, and C. Zheng. Understanding user's query intent with wikipedia. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 471–480, New York, NY, USA, 2009. ACM.

[6] D. Shen, R. Pan, J.-T. Sun, J. J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352, 2006.