Mining Relevant Text from Unlabelled Documents

Daniel Barbará Carlotta Domeniconi Ning Kang
Information and Software Engineering Department
George Mason University
Fairfax, VA 22030
{dbarbara,cdomenic,nkang}@gmu.edu

Abstract

Automatic classification of documents is an important area of research with many applications in the fields of document searching, forensics and others. Methods to perform classification of text rely on the existence of a sample of documents whose class labels are known. However, in many situations, obtaining this sample may not be an easy (or even possible) task. In this paper we focus on the classification of unlabelled documents into two classes: relevant and irrelevant, given a topic of interest. By dividing the set of documents into buckets (for instance, answers returned by different search engines), and using association rule mining to find common sets of words among the buckets, we can efficiently obtain a sample of documents that has a large percentage of relevant ones. This sample can be used to train models to classify the entire set of documents. We prove, via experimentation, that our method is capable of filtering relevant documents even in adverse conditions where the percentage of irrelevant documents in the buckets is relatively high.

1. Introduction

In information retrieval, such as content based image retrieval or web page classification, we face an asymmetry between positive and negative examples [10, 2]. Suppose, for example, we submit a query to multiple search engines. Each engine retrieves a collection of documents in response to our query. Such collections include, in general, both relevant and irrelevant documents. Suppose we want to discriminate the relevant documents from the irrelevant ones. The set of all relevant documents in all retrieved collections represent a sample of the positive class, drawn from an underlying unknown distribution. On the other hand, the irrelevant documents may come from an unknown number of different "negative" classes. In general, we cannot approximate the distributions of the negative classes, as we may have too few representatives for each of them. Hence, we are facing a problem with an unknown number of classes, with the user interested in only one of them.

Modelling the above problem as a two-class problem, may impose misleading requirements, that can yield poor results. We are definitely better off focusing on the class of interest, as positive examples in this scenario have a more compact support, that reflects the correlations among their feature values.

Moreover, more often than not, the class labels of the data are unknown, either because the data is too large for an expert to label it, or because no such expert exists. In this work we eliminate the assumption of having even partially labelled data. We focus on document retrieval, and develop a technique to mining relevant text from unlabelled documents. Specifically, our objective is to identify a sample of positive documents, representative of the underlying class distribution. The scenario of a query submitted to multiple search engines will serve as running example throughout the paper, although the technique can be applied to a variety of scenarios and data. Our approach reflects the asymmetry between positive and negative data, and does not make any particular and unnecessary assumption on the negative examples.

2 Related Work

In [4] the authors discuss a hierarchical document clustering approach using frequent set of words. Their objective is to construct a hierarchy of documents for browsing at increasing levels of specificity of topics.

In [1] the authors consider the problem of enhancing the performance of a learning algorithm allowing a set of unlabelled data augment a small set of labelled examples. The driving application is the classification of Web pages. Although similar to our scenario, the technique depends on the existence of labelled data to begin with.

The authors in [6] exploit semantic similarity between terms and documents in an unsupervised fashion. Docu-

ments that share terms that are different, but semantically related, will be considered as unrelated when text documents are represented as a *bag of words*. The purpose of the work in [6] is to overcome this limitation by learning a *semantic proximity matrix* from a given corpus of documents by taking into consideration high order correlations. Two methods (both yielding to the definition of a kernel function) are discussed. In particular, in one model, documents with highly correlated words are considered as having similar content. Similarly, words contained in correlated documents are viewed as semantically related.

3 The DocMine Algorithm

Given a document, it is possible to associate with it a *bag of words* [5, 3, 7]. Specifically, we represent a document as a binary vector $\mathbf{d} \in \Re^n$, in which each entry records if a particular word stem occurs in the text. The dimensionality n of $\mathbf{d}$ is determined by the number of different terms in the corpus of documents (size of the *dictionary*), and each entry is indexed by a specific term.

Going back to our example, suppose we submit a query to s different search engines. We obtain s collections, or *buckets*, of documents $B_j = \{\mathbf{d}_i\}_j, \quad j = 1, \ldots, s$.

While many documents retrieved by a specific search engine (a bad one) might be irrelevant, the relevant ones are expected to be more frequent in the majority of buckets. In addition, since we can assume that positive documents are drawn from a single underlying distribution, a compact support unifies them across all buckets. On the other hand, the negatives manifest a large variation. We make use of these characteristics to develop a technique that discriminates relevant documents from the irrelevant ones. In details, we proceed as follows.

We mine each bucket B_j to find the frequent itemsets that satisfy a given support level. Each resulting itemset is a set of words. The result of this process is a collection of sets of itemsets, one set for each bucket: $F_j = \{W_i | W_i$ *is a frequent itemset in bucket* $j\}$ for $j = 1, \ldots, s$, where it is possible that $F_j = \emptyset$, for some j. Now we compute all itemsets that are frequent in m buckets: $I_m = \{W_i | W_i \in F_{j_1} \cap F_{j_2} \cap \ldots \cap F_{j_m}\}$, for distinct $j_1, \ldots, j_m$. In general $m = \lfloor s/2 \rfloor + 1$. In our experiments we set $m = s$ since we consider a limited number of buckets ($s = 5$), driven by the number of available documents per topic. We wish now to retrieve the documents that *support* the itemsets that are frequent in m buckets. Then, for each $W_i \in I_m$, we select, in each of the m buckets that contain W_i as frequent itemset, the documents that has W_i expressed within. The resulting collection of documents P represent the presumed positive documents, relevant to our query.

The algorithm, which we call DocMine (**Doc**ument **Min**ing), is summarized in the following. The algorithm takes as input the s buckets of documents, and the minimum support (Sup_{min}) for the computation of frequent itemsets.

1. **Input**: s buckets of documents $B_j = \{\mathbf{d}_i\}_j$, $j = 1, \ldots, s$, Sup_{min}, m.
2. Compute frequent itemsets in each bucket B_j:

   $F_j = \{W_i | W_i \; is \; a \; frequent \; itemset \; in \; bucket \; j\}$, $j = 1 \ldots, s$
3. Compute all itemsets that are frequent in m buckets:

   $I_m = \{W_i | W_i \in F_{j_1} \cap F_{j_2} \cap \ldots \cap F_{j_m}\}$
4. Set $P = \emptyset$.
5. for each $W_i \in I_m$
   - for each $l = 1, \ldots, m$ such that $W_i \in \cap_{l=1}^{m} F_{j_l}$
     - for each $\mathbf{d} \in B_{j_l}$
       - if $\mathbf{d}$ contains W_i
         - $P = P \cup \{\mathbf{d}\}$
6. **Output**: the set P (presumed positive documents)

It is important to remark that the DocMine algorithm can be tuned to ignore itemsets of small size. Some words, in fact, may be common to documents of different topics (they would not discriminate). Our experience tells us that, for instance, combinations of two frequent words are not sufficient to discriminate among different topics.

4 Experimental Results

To test the feasibility of our approach we use the Reuters-21578 text categorization collection [8], omitting empty documents and those without labels. Common and rare words are removed, and the vocabulary is stemmed with the Porter Stemmer [9]. After stemming, the vocabulary size is 12113.

In our experiments, we consider five buckets of documents ($s = 5$), and vary the percentage R of relevant documents (i.e., concerning the topic of interest) in each bucket from 50% to 80%. As topics of interest, we select the topics with the largest number of documents available in the data set. Once we have identified a topic, the non relevant documents are randomly selected from the remaining topics. We observe that some documents in the Reuters data have multiple topics associated (e.g., *grain* and *crops*). In our experiments, a document is considered positive if it has the topic of interest among its associated topics. For each topic examined, we test three different values of the minimum support (10%, 5%, 3%).

We have also investigated different threshold values (from 2 to 5) for the cardinality of the frequent itemsets ($|W_i|$). Only frequent itemsets of size above (or equal to)

the threshold are considered for the retrieval of relevant documents. The rationale beyond this test is that if an item is too common across different documents, then it would have little discriminating power. The setting of a proper threshold for $|W_i|$ allows to discard frequently used words (not removed during preprocessing) that are not discriminating. Our experiments show that threshold values of 4 or 5 (depending on the value of the minimum support) give good results.

In the following tables we report, for each value of R, the number of (retrieved) documents in P ($|P|$), the number of positive (relevant) documents in P ($|P^+|$), the percentage of positive documents in P ($\%|P^+|$) –precision –, and the percentage of positive documents retrieved by P (r) –recall–. Each caption has (in parenthesis) the total number of positive documents versus the total number of documents in the five buckets.

We have considered different topics in our experiments. For lack of space, we report only the results for the topic *earn* (3776 documents). Similar results were obtained for the other topics. We distribute all the available positives among the buckets, and adjust the number of negatives accordingly to the R value considered.

Tables 1-4 show the results. Figures 1-3 plot the precision values for the topic *earn*, for increasing threshold t on the itemset size $|W_i|$. Each line corresponds to a value of R (percentage of positive documents in each bucket). The plots show that, in each case, the setting of $t = 5$ allows the achievement of a precision value very close to 1. For larger support values (5% and 10%), $t = 4$ suffices for the selection of an almost "pure" sample of documents. Even in the adverse condition of 50% of irrelevent documents in the buckets, the DocMine algorithm is able to achieve a very high precision.

These results are very promising for the purpose of constructing a classifier that uses the selected collection of documents P as a positive sample.

5 Conclusions

We have introduced a new algorithm, based on association rule mining, to select a representative sample of positive examples from a given set of unlabelled documents. Our experiments show that our method is capable of selecting sets of documents with precision above 90% in most cases, when frequent itemsets of cardinality 4 or 5 are considered. We emphasize that, in all cases, the precision tends to reach high levels, as the cardinality of the common itemsets grows, regardless of the value of the support, or the percentage of relevant documents in the original buckets.

Table 1. Topic *earn*: $R = 50\%$ ($3776/7552$).

Sup_{min}	$\|W_i\|$	$\|P\|$	$\|P^+\|$	$\%\|P^+\|$	r
10%	≥ 2	5323	2824	0.53	0.74
	≥ 3	2538	2204	0.87	0.58
	≥ 4	1848	1848	1.00	0.49
	≥ 5	1103	1103	1.00	0.29
5%	≥ 2	6441	3012	0.47	0.80
	≥ 3	4653	2597	0.56	0.69
	≥ 4	1972	1913	0.97	0.51
	≥ 5	1284	1284	1.00	0.34
3%	≥ 2	7246	3597	0.50	0.95
	≥ 3	5789	2943	0.51	0.78
	≥ 4	3671	2408	0.66	0.64
	≥ 5	1642	1628	0.99	0.43

Table 2. Topic *earn*: $R = 60\%$ ($3776/6294$).

Sup_{min}	$\|W_i\|$	$\|P\|$	$\|P^+\|$	$\%\|P^+\|$	r
10%	≥ 2	4453	2932	0.66	0.78
	≥ 3	2725	2250	0.83	0.60
	≥ 4	1842	1841	0.99	0.49
	≥ 5	1403	1403	1.00	0.37
5%	≥ 2	5684	3507	0.62	0.93
	≥ 3	3985	2668	0.67	0.71
	≥ 4	2045	1999	0.98	0.53
	≥ 5	1381	1376	0.99	0.36
3%	≥ 2	5859	3561	0.61	0.94
	≥ 3	4636	2928	0.63	0.78
	≥ 4	3311	2490	0.75	0.66
	≥ 5	1879	1875	0.99	0.50

Table 3. Topic *earn*. $R = 70\%$ ($3776/5394$).

Sup_{min}	$\|W_i\|$	$\|P\|$	$\|P^+\|$	$\%\|P^+\|$	r
10%	≥ 2	3592	2940	0.82	0.78
	≥ 3	2515	2274	0.90	0.60
	≥ 4	1849	1842	0.99	0.49
	≥ 5	1674	1674	1.00	0.44
5%	≥ 2	4784	3467	0.72	0.92
	≥ 3	3253	2747	0.84	0.73
	≥ 4	2027	1989	0.98	0.53
	≥ 5	1644	1642	0.99	0.43
3%	≥ 2	4982	3555	0.71	0.94
	≥ 3	4422	3447	0.78	0.91
	≥ 4	3550	3079	0.87	0.81
	≥ 5	1807	1803	0.99	0.48

References

[1] Blum, A., Mitchell T. (1998). Combining Labelled and Unlabelled Data with Co-Training. *Proceedings of the 1998 Conference on Computational Learning Theory*.

[2] Chen, Y., Zhou, X. S., Huang, T. S. (2001). One-class SVM for learning in image retrieval. *Proceedings of International Conference on Image Processing*.

[3] Dumais, S. T., Letsche, T. A., Littman, M. L., & Landauer, T. K. (1997). Automatic cross-language retrieval using latent semantic indexing. *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*.

[4] Fung, B. C. M., Wang, K., & Ester M. (2003). Hierarchical Document Clustering Using Frequent Itemsets. *Proceedings of the SIAM International Conference on Data Mining*.

[5] Joachims, T. (1998). Text categorization with support vector machines. *Proceedings of European Conference on Machine Learning*.

[6] Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002). Learning Semantic Similarity. *Neural Information Processing Systems (NIPS)*.

[7] Leopold, E., & Kindermann, J. (2002). Text categorization with support vector machines, how to represent texts in input space? *Machine Learning*,**46**,423-444.

[8] Lewis, D., Reuters-21578 Text Categorization Test Collection Distribution 1.0. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[9] Porter, M. (1980). An algorithm for suffix stripping, Program, **14**(3): 130-137 http://www.tartarus.org/∼martin/PorterStemmer

[10] Zhou, X. S., & Huang, T. S. (2001). Small sample learning during multimedia retrieval using BiasMap. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

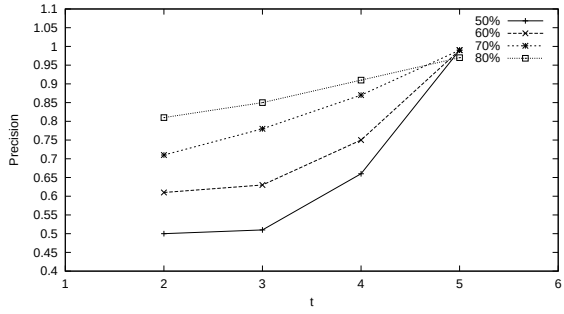

Figure 1. Precision values for topic *earn* **and $Sup_{min} = 3\%$. The x-axis is the minimum cardinality of common itemsets (t).**

Table 4. Topic *earn***: $R = 80\%$ ($3776/4720$).**

Sup_{min}	$\|W_i\|$	$\|P\|$	$\|P^+\|$	$\%\|P^+\|$	r
10%	≥ 2	3192	2810	0.88	0.74
	≥ 3	2398	2279	0.95	0.60
	≥ 4	1394	1393	0.99	0.37
	≥ 5	1205	1205	1.00	0.32
5%	≥ 2	4151	3483	0.84	0.92
	≥ 3	3003	2763	0.92	0.73
	≥ 4	2126	2111	0.99	0.56
	≥ 5	1589	1587	0.99	0.42
3%	≥ 2	4294	3493	0.81	0.93
	≥ 3	3854	3275	0.85	0.87
	≥ 4	3059	2780	0.91	0.74
	≥ 5	2447	2377	0.97	0.63

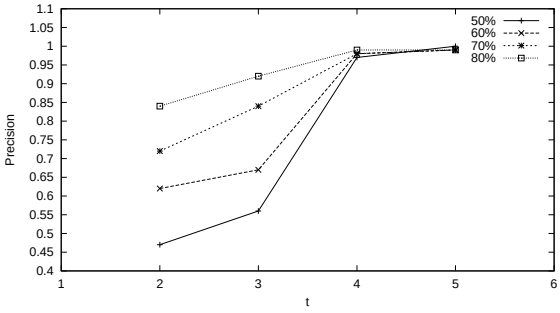

Figure 2. Precision values for topic *earn* **and $Sup_{min} = 5\%$. The x-axis is the minimum cardinality of common itemsets (t).**

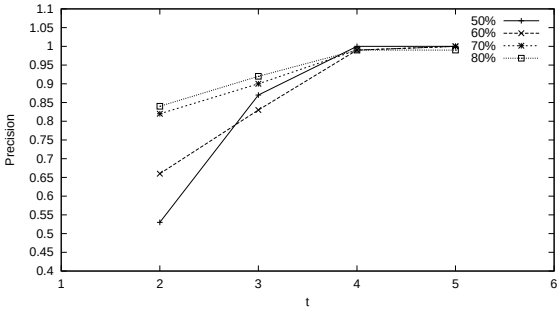

Figure 3. Precision values for topic *earn* **and $Sup_{min} = 10\%$.The x-axis is the minimum cardinality of common itemsets (t).**