SCALABILITY OF TEXT CLASSIFICATION

Ioannis Antonellis, Christos Bouras, Vassilis Poulopoulos, Anastasios Zouzias
Research Academic Computer Technology Institute, N. Kazantzaki, University Campus,
GR-26500, Rio Patras, Greece
and
Computer Engineering and Informatics Department,
University of Patras
GR-26500, Patras, Greece

Keywords: Scalable text classification, scalability, text decomposition, term-by-sentences matrix, web personalization.

Abstract: We explore scalability issues of the text classification problem where using (multi)labeled training documents we try to build classifiers that assign documents into classes permitting classification in multiple classes. A new class of classification problems, called 'scalable' is introduced that models many problems from the area of Web mining. The property of scalability is defined as the ability of a classifier to adjust classification results on a 'per-user' basis. Furthermore, we investigate on different ways to interpret personalization of classification results by analyzing well known text datasets and exploring existent classifiers. We present solutions for the scalable classification problem based on standard classification techniques and present an algorithm that relies on the semantic analysis using document decomposition into its sentences. Experimental results concerning the scalability property and the performance of these algorithms are provided using the 20newsgroup dataset and a dataset consisting of web news.

1 INTRODUCTION

Text classification (categorization) is the procedure of assigning a category label to documents. In tradition, decision about the label assignment is based on information gained by using a set of pre-classified text documents in order to build the classification function. So far, many different classification techniques have been proposed by researchers, e.g. naïve Bayesian method, support vector machines (SVM), Rocchio classifier (Rocchio, 1971) (vector space representation), decision trees, neural networks and many others (Yang et al., 1999).

However, depending on the selection of specific parameters of classification procedure, there exist different variations of the problem. Concerning training data, we can have labeled data for all existing categories or only positive and unlabeled examples. Training documents can also be multi-labeled, that is some documents may have been assigned many labels. Correspondingly, classification of new documents may vary from the assignment of a simple category label per document to many different labels as we can permit multi-label classification. Finally, definition of the categories may be statically initialized from the set of labels that training documents define, or we may want to define new categories 'on-the-fly' or even delete some others.

Text classification procedures can find applications on many different research areas. Traditionally, text segmentation and summarization techniques share a lot with text categorization, as well as recent advances (Kumatan et al., 2004) in topic event detection techniques (TDT) indicate that performance of new event detection (NED) can be improved by the use of text classification techniques. Standard text classifiers are also the kernel of many web-mining techniques that mostly deal with structured or semi-structured text data. In this case, classifiers are further enhanced in order to exploit information about the structure of the documents and refine results.

In this paper, we introduce a new class of classification problems, called scalable, that can be seen as a formal definition of different, existing classification problems under a unified, general formalism. However, it addresses new issues in the classification procedure, such as the definition of different similarity classes. Such an approach, can

Antonellis I., Bouras C., Poulopoulos V. and Zouzias A. (2006).
SCALABILITY OF TEXT CLASSIFICATION.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 408-413
DOI: 10.5220/0001240904080413

properly formalize many classification problems that derive from web mining problems such as page ranking algorithms, personalization of search results and many others; for possible applications see (Antonellis et al. 2005). Although, we can build trivial solutions for this problem using existing classification techniques, we study a specific technique that exploits the semantic information that derives from the decomposition of training documents into their sentences. Such a semantic analysis shares a lot with passage-based retrieval techniques (Salton et al, 1996), (Hearst et al., 1993) that further decompose a large text document in order to identify text segments and text themes so as to improve accuracy of information retrieval queries. It is also connected with already proposed phrase-based document indexing techniques (Hammouda et al., 2004) as an alternative to single-word analysis that the simple Vector Space Model provides.

The rest of paper is organized as follows. In Section 2 the definition of the Scalable Classification Problem is presented, along with an intuitive description of possible applications. In Section 3, we study the 20newsgroup dataset applying a new text analysis technique so as to specify logical interpretation of the 'user-specific' classification. Subsequently, different solutions for the problem are described that base upon the reduction of the problem into multiple standard binary classification problems. Section 5 describes our Scalable Classification Algorithm that derives from spectral decomposition of the training documents into the vector representation of their sentences. Experimental evaluation of the algorithm is given in Section 6, using two different datasets. Finally, we summarize our results and address future work.

2 SCALABLE TEXT CLASSIFICATION PROBLEM

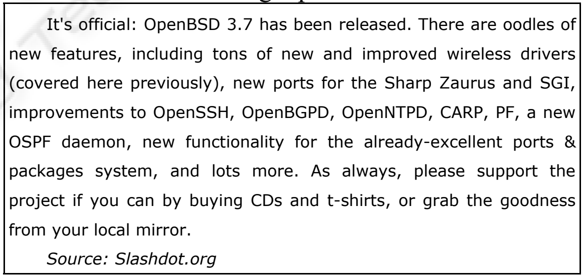
Traditional text classification is the task of assigning a Boolean value to each pair $\langle d_j, c_i \rangle \in D \times C$, where D is a domain of documents and $C = \{c_1, \ldots, c_{|C|}\}$ is a set of predefined categories. More formally, we have the following definition (Sebastiani, 2002):

DEFINITION 1 (STANDARD TEXT CLASSIFICATION) Let $C = \{c_1 \ , \ldots, c_{|C|}\}$ be a set of predefined categories and $D = \{d_1, \ldots, d_{|D|}\}$ a growing set of documents. A standard text classifier is an approximation function $\breve{\Phi} = D \times C \to \Re$ of the function $\Phi = D \times C \to \Re$ that describes how documents ought to be classified.

Looking further into the definition, it is easy to see that most parameters of the problem are static. Definition of the categories relies only on the initial set that training, labeled documents specify and cannot be further expanded or limited. Moreover, definition of a specific category relies only on information that training documents provide. Classification function is specified by the minimization of an effectiveness measure (Sebastiani, 2002) that shows how much functions $\breve{\Phi}$ and Φ 'coincide'. In tradition, this measure relies on the precision and recall, or other effectiveness measures that combine these values (e.g. micro-averaging and macro-averaging). It is then obvious that depending on the measure we choose, resulting classifiers defer. However, we can argue that classification procedure still remains static, that is, given a classifier and a specific document, whenever we try to apply the classifier to that document, classification result will remain the same (by definition).

Web mining techniques that capture user-profile information in order to improve end-user results, many times come up with text classification problems. However, characteristics of these text classification problems involve dynamic changes of Web users' behaviour and 'on-the-fly' definition of the category topics.

Consider, for example, the text article of Figure 1 and Web users A and B. A is a journalist that looks for information about Linux in order to write an article about open source software in general, while B is an experienced system administrator looking instructions on installing OpenBSD 3.6.



It's official: OpenBSD 3.7 has been released. There are oodles of new features, including tons of new and improved wireless drivers (covered here previously), new ports for the Sharp Zaurus and SGI, improvements to OpenSSH, OpenBGPD, OpenNTPD, CARP, PF, a new OSPF daemon, new functionality for the already-excellent ports & packages system, and lots more. As always, please support the project if you can by buying CDs and t-shirts, or grab the goodness from your local mirror.

Source: Slashdot.org

Figure 1: Example news article.

A well-trained standard classification system would then provide the above document to both users, as it is clearly related to open source software and to OpenBSD operating system. However, it is obvious that although user A would need such a decision, it is useless for user B to come across this article.

Trying to investigate the cause of user's B disappointment, we see that standard text classification systems lack the ability to provide 'per-user' results. However, a user's knowledge of a topic should be taken into account while providing him with the results. It is more possible that a user who is aware of a category (e.g. user B knows a lot about Linux) would need less and more precise results, while non-expert users (such as the journalist) will be satisfied with a variety of results.

Scalable text classification problem can be seen as a variant of classical classification where many similarity classes are introduced and permit different, multi-label classification results depending on the similarity class.

DEFINITION 2 (SCALABLE TEXT CLASSIFICATION) Let $C = \{c_1 ,\ldots, c_{|C|}\}$ a set of growing set of categories and $D = \{d_1, \ldots, d_{|D|}\}$ a growing set of documents. A scalable text classifier is a function $\Phi = D \times C \to \Re^p$ that introduces p different similarity classes.

It follows from Definition 2 that given an initial test set of k training data (text documents) TrD = {trd$_1$, trd$_2$, …, trd$_k$} already classified into m specific, training categories from a well-defined domain TrC = {trc$_1$, trc$_2$, …, trc$_m$}, the scalable text classifier is a function that not only maps new text documents to a member of the TrC set using the training data information but also: (a) defines p similarity classes and p corresponding similarity functions that map a document into a specific category c. Similarity classes can be seen as different ways to interpret the general meaning (concept) of a text document. (b) Permits the classification of each document into different categories depending on the similarity class that is used. (c) Permits the definition of new members and the erasure of existing ones from the categories set. That means that the initial set TrC could be transformed into a newly defined set C with or without all the original members, as well as new ones.

3 EXPLORING SCALABILITY ON TEXT DATASETS

When trying to interpret 'per-user' classification results as a way to provide user with results that match his expertise-familiarity of a topic, we need to be full aware of the topics that a document includes. However, using the simple vector space representation of the text, the more we can do is treat each term as a different topic resulting on a Boolean-like schema. Below, we study a specific decomposition of text documents that enables us identify further subtopics of a document. The document vector is decomposed into the vector representation of its sentences, revealing further subtopics. In fact, using the 20newsgroup dataset we used this decomposition to compute the cosine similarities of each sentence of a document with the different category vectors of the dataset. The corresponding results prove that this technique can be used in order to construct scalable classifiers. Scalability of these classifiers can be achieved by varying the number of sentence vectors that we demand to be close to the category vector.

We study decomposition of document vectors into further components. Having the vector space representation of a document, it is clear that we have no information on how such a vector has been constructed, as it can be decomposed in unlimited ways into a number of components. Therefore we lose information regarding the subtopics and the structure of the documents. However, property of scalability demands to have a picture of the subtopics that a document includes. As an alternative, we propose to decompose every document into the components that represent its sentences and use this decomposition while making decision on the classification. We therefore have the following definition of the document decomposition into its sentences:

DEFINITION 3 (DOCUMENT DECOMPOSITION INTO SENTENCES) Let $\vec{d}_i = [v_1, v_2, \ldots, v_k]$ the vector representation of a document d_i. A document decomposition into its sentences is a decomposition of vector $\vec{d}_i$ of the form $\vec{d}_i = \vec{s}_1 + \vec{s}_2 + \ldots + \vec{s}_n$, where component s_k is a vector $\vec{s}_k = [v'_1, v'_2, \ldots, v'_{|s_k|}]$ representing k-th sentence of document d_i

To explore document decomposition into sentences we used the 20 newsgroup dataset, a collection of articles of 20 newsgroups. Each category contains 1000 articles. In order to evaluate the similarity values between different category vectors we used the standard cosine metric (Jones et al. 1987) Using this dataset, we computed for each category the cosine similarities of the sentence vectors with the category vector. Figure 3 presents these results for three different categories. The basic results can be summarized as:

- Categories with general topic (like alt.atheism or soc.religion.christian) have a dense uniform allocation of similarities in the range [0-0.1] and a sparse uniform allocation in the range [0.1 – 0.5].
- Well structured categories seem to be indicated from a uniform sentence vs. category similarity chart.

Trying to investigate on an easy way to identify general categories, non-well structured categories seem to reside on 'term to sentence' matrices that have a blocked structure. Figure 3 provides a visualization of the matrix elements of the 'term to sentence' matrix where large values are identified by intense color. Figures of categories that were identified as not well structured using the similarity chart are shown to have a matrix with blocked structure (e.g. (d) or (e) matrices).

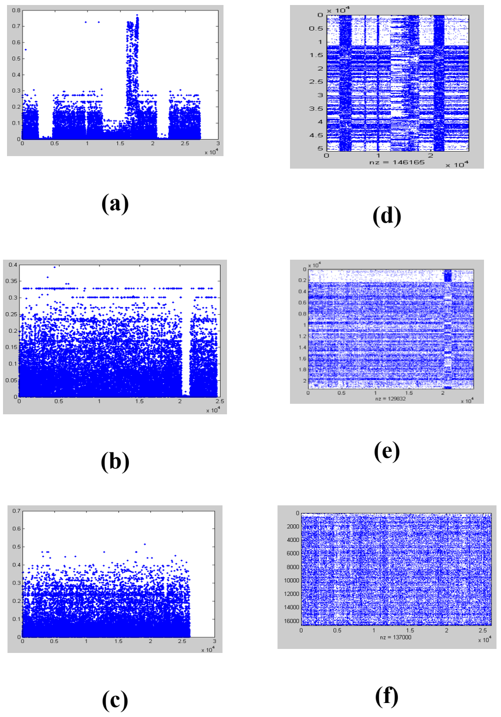


Figure 3: Sentence vs category vectors for different categories of the 20-newsgroup dataset (first line) with the corresponding 'term-to-sentences' matrix using function spy of MATLAB (right column) (a) comp.os.ms-windows.misc (b) comp.windows.x (c) talk.politics.misc.

4 SOLUTIONS BASED ON STANDARD CLASSIFIERS

There are two main alternative approaches to multi-label classification problem using existing standard classification techniques. The first is to build a binary classifier that recognizes each class (resulting in a classifier per class) (Yang, 1999), (Nigam et al, 2000). The second is to correlate each class – document pair with a real value score, and use the resulting scores in order to rank the relevance of a document with each class. Classes that match some threshold criterion can then be assigned to the document.

Below we present modified versions of standard text classification techniques that permit definition of many similarity classes and therefore they can be considered as solutions of the scalable classification problem. Multi-labeled results are obtained by following the above-mentioned first technique that is the construction of many binary classifiers (one for each category). In addition, each classifier defines a scalability function that is a function that we can adjust in order to tune the similarity class.

4.1 Scalable Naïve Bayes

Naïve Bayes classifier treats a document as a vector of attributes. In order to permit introduction of different similarity classes we can rank categories depending on a-posteriori probability. The scalability function is then the a-posteriori probability and we define similarity classes that select the category with a specific rank position. We define that i-similarity class selects the category that its a posteriori probability has rank i.

4.2 Scalable Rocchio Classifier

Rocchio is an early text classification method (Rocchio, 1971). In this method, each document is represented as a vector, and each feature value in the vector is computed using classic TD-IDF scheme (Salton et al, 1983). Different similarity classes can be easily constructed by ranking categories according to the cosine similarity (scalability function) of the document and the categories vectors. Again, categories are ranked in increasing order and i-similarity class selects the category that its vector's cosine has rank i.

4.3 Scalable k Nearest Neighbors

Given a test document, the kNN algorithm finds the k nearest neighbors among the training documents, and uses the categories of the k neighbors to weight the category candidates. The similarity score of each neighbor document to the test document is used as the weight of the categories of the neighbor document. If several of the k nearest neighbors share a category, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of candidate categories, a ranked list can be obtained for the test document. Definition of the similarity classes can be obtained by using the ranked list of the categories sorted by the scores, so as i-similarity class selects the i-th category as the classification result. The scalability function is the score value (weighted sum).

5 SCALABLE CLASSIFICATION ALGORITHM

The algorithm we present here requires an initial set of predefined categories and their corresponding labeled data. The most useful characteristic of the proposed classification algorithm is its scalability feature. A text document can be classified into many different categories depending on the similarity of the semantic representation of its sentences with the categories. Exploiting user's level of expertise in a specific area, we can relax or tighten the specific number of sentences that we demand to match a similarity threshold, in order to allow classification of the article in many categories. Formal definition of the Training Phase (in MATLAB pseudocode) of the Scalable classification algorithm is shown in Figure 4:

```
Algorithm Training Phase (Text of Document $d_i$)

1. Decompose labeled text documents into their sentences
2. Compute term to sentences matrix of every category using some
   indexing method
3. Compute category vectors by combining the columns of the
   corresponding term to sentences matrix
4. Estimate categories similarity threshold, by computing the cosines of
   the angles between the different category vectors of step 3
5. For each category, estimate sentences similarity threshold by
   computing the cosines of the angles between all sentence vectors
   with the corresponding category vector
```

Figure 4: Training Phase of the Scalable Classification Algorithm.

Main characteristics of the classification phase (Figure 5) include the ability to adjust the number of sentences k that must match a sentences similarity threshold in order to classify the corresponding document to a category and the feedback that the algorithm implicitly takes in order to re-compute categories vectors and therefore capture semantic changes of the meaning of a topic as time (arrival of new text documents) passes.

```
Algorithm [category, multicategory]=Classification Phase (Text of
Document $d_i$)

1. Decompose unlabeled text document into its sentences
2. Compute term to sentences matrix of the document. Let $d_i = [s_1, s_2, …, s_n]$ be the term-to-sentences matrix of document $d_i$.
3. Compute document vector by combining the columns of the term to
sentences matrix $d_i = sum(s_i)$, i = 1, …, n
4. Estimate similarity (cosine) of the document vector with the category
vectors computed at step 3 of Training Phase. If cosine matches a similarity
threshold computed at step 4 of Training Phase classify the document to the
corresponding category. Let $c_j$ the category vector and $t_j$ the threshold of $c_j$,
and k the number of the category vectors.
    Category=-1;
    for j=1 to k do
        if ($t_j$<cos($d_i$, $c_j$)) do
            category = j;
            goto step 7;
        end if
    End for
5. Estimate similarity (cosines) of each sentence with the category
vectors computed at step 3 of Training Phase
6. If a cosine matches a similarity threshold computed at step 5 of
Training Phase classify the document to the corresponding category
(allowing scalable multi-category document classification). Let $c_j$ the
category vector and $ts_j$ the sentence threshold of $c_j$, k be the number of the
category vectors.
7. for i=1 to n do
8. multicategory=[];
        for j=1 to k do
            if ($ts_j$<cos($s_j$, $c_j$)) do
                multicategory =[ multicategory j];
                continue;
            end if
        end for
9. end for
10. category=category+normalize($d_i$);
```

Figure 5: Classification Phase of the Scalable Classification Algorithm.

6 EXPERIMENTS

In this section we provide experimental evaluation of the scalability property of the Scalable classification algorithm we presented. We define scalability property as the ability of a classifier to adjust classification results in a 'per-user' basis. In order to measure scalability, we measured the total number of documents that were returned to the output of the classifiers for different values of the scalability function of each classifier. For the evaluation we used the 20 newsgroup dataset and a news dataset consisting of five general categories (business, education, entertainment, health, and politics) with articles from different well known news portals. All experiments were conducted using both the Rainbow tool as well as TMG (Zeimpekis and Gallopoulos, 2005).

Table 1: Average F-scores for different number of similarity classes.

	p	Average F score
20-newsgroup	2	0,83
	3	0,79
	4	0,79
	5	0,71
Web news	2	0,91
	3	0,86
	4	0,87
	5	0,79

Evaluation of the accuracy of the algorithm can be seen on Table 1, where average F-scores are presented for different numbers of total similarity classes (p).

7 CONCLUSIONS AND FUTURE WORK

We see two main achievements in this paper. Firstly, scalability issues of text classification problem were studied resulting in a formal definition of a wide range of new classification problems. Definition of different similarity classes introduces a new way to represent formally the need for 'per-user' results that a large range of applications demand. In addition, representation of categories using category vectors permits the use of feedback acquired by newly classified text documents in order to re-define categories. Such an approach results in following a topic's meaning while time passes and capturing semantic changes. Besides, a text analysis technique based on document decomposition into its sentences was presented and applied into the scalable classification problem resulting in an efficient algorithm. To the best of our knowledge, such an approach is the first text processing technique to exploit the lack of certainty of a user's information need that different applications imply in order to relax or tighten a similarity threshold and provide users with a wider or tighter set of answers. As experimental analysis proved, this technique provides a powerful tool for the analysis of text datasets, the identification of abnormalities as well as provides very accurate results for different number of similarity classes.

Future work will include further exploration of the presented text analysis technique and direct use of it for web mining problems. There is also need for development of well-specified datasets for the evaluation of future algorithms on the scalable classification problem.

ACKNOWLEDGMENTS

Ioannis Antonellis's work was partially supported by a Zosima Foundation Scholarship under grant 1039944/ 891/ B0011 /21-04-2003 (joint decision of Ministry of Education and Ministry of Economics).

REFERENCES

I. Antonellis, C. Bouras, V. Poulopoulos, *"Personalized News Categorization through Scalable Text Classification"*, In Proc. of 8th Asia Pacific Web Conference (APWEB 2006), pp. 391-401 (to appear)

F. Sebastiani, "*Machine Learning in automated text categorization*", ACM Comput. Surv 2002., Vol 34, No 1, pp 1 - 47

G. Kumaran and J. Allan, "*Text classification and named entities for new event detection*", SIGIR '04

Y. Yang, "*An evaluation of statistical approaches to text categorization*", J. Information Retrieval, Vol 1, No. 1/2, pp 67-88, 1999

K. Nigam, A. McCallum, S. Thrun and T. Mitchell. "*Text Classification from Labeled and Unlabeled Documents using EM*". Machine Learning, 39(2/3). pp. 103-134. 2000

J. Rocchio, Relevant feedback in information retrieval.. In G. Salton (ed.). "*The smart retrieval system-experiments in automatic document processing*", 1971, Englewood Cliffs, NJ.

Salton, G. and McGill, M. (1983). "*Introduction to Modern Information Retrieval*". McGraw-Hill.

Buckley, C., Salton G., and Allan J. (1994)."*The effect of adding relevance information in a relevance feedback environment*", SIGIR-94.

W. Jones and G. Furnas, Pictuers of relevance: "*A geometric analysis of similarity measures*", J. American Society for Information Science, 38 (1987), pp. 420-442

D. Zeimpekis, E. Gallopoulos, "*Design of a MATLAB toolbox for term-document matrix generation*", Proceedings of the Workshop on Clustering High Dimensional Data, SIAM 2005 (to appear)

Y. Yang and X. Liu, "*A re-examination of text categorization methods*", Proceedings of ACM SIGIR 1999, pp 42-49

G. Salton, A. Singhal, C. Buckley, M. Mitra, "Automatic Text Decomposition Using Text Segments and Text Themes", in Proc. 7th ACM Conf. Hypertext, Washington, DC, Mar. 1996, pp. 53-65

K. Hammouda, K. Mohamed, "Efficient Phrase-Based Document Indexing for Web Document Clustering", *IEEE Transactions on Knowledge and Data Engineering* 16, 10 (Oct. 2004), 1279-1296

M. A. Hearst and C. Plaunt. "*Subtopic structuring for full-length document access",* SIGIR 1993, Pittsburgh, PA, pp 59-68, 1993