Scalable Text Classification as a tool for Personalization

Ioannis Antonellis, Christos Bouras, Vassilis Poulopoulos

Research Academic Computer Technology Institute N. Kazantzaki, University Campus, GR-26500 Patras, Greece

and

Computer Engineering and Informatics Department., University of Patras, GR-26500 Patras Greece

antonell@ceid.upatras.gr, bouras@ceid.upatras.gr, poulop@ceid.upatras.gr

Contact Person:

Christos Bouras

Research Academic Computer Technology Institute, N. Kazantzaki, GR-26500 Patras, Greece
and
Computer Engineering and Informatics Department., Univ. of Patras, GR-26500 Patras, Greece

Tel:+30-(2)610-960375

Fax:+30-(2)610-969016

e-mail: bouras@ceid.upatras.gr

Scalable Text Classification as a tool for Personalization

Abstract—**We consider scalability issues of the text classification problem where by 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, with applications on web mining. Scalable classification utilizes newly classified instances in order to improve the accuracy of future classifications and capture changes in semantic representation of different topics. In addition, definition of different similarity classes is allowed, resulting in a 'per-user' classification procedure. Such an approach provides a new methodology for building personalized applications. This is due to the fact that the user becomes a part of the classification procedure. We explore solutions for the scalable text classification problem and introduce an algorithm that exploits a new text analysis technique that decomposes documents into the vector representation of their sentences according to the user expertise. Finally, a web-based personalized news categorization system that bases upon this approach is presented.**

Index Terms

8.3 Data Mining – Web–based information

14.1 Information Retrieval – Customization and user profiles

30.1 Web – Inf. Services on the Web

I. 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 [20] (vector space representation), decision trees, neural networks and many others [13]

Depending on the selection of specific parameters of the 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 procedure can find applications on many different research areas. In tradition, text segmentation and summarization techniques share a lot with text categorization, as well as recent advances [14]. 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 properly formalize many classification problems that derive from web mining problems such as page ranking algorithms, personalization of search results and many other. 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.

We focus on the needs of the Internet users who access news information from major or minor news portals. From a very brief search we found more than thirty portals that exist only in USA. This means that if one wants to find information regarding to a specific topic, (s)he will have to search one by one, at least the major portals, and try to find the news of his preference. A better solution is to access every site and search for a specific topic if a search field exists in the portals. The problem becomes bigger for someone who would like to track a specific topic daily (or more times per day). This means that the users have to visit every single site and try to search for their topic, which is a tradition for the internet user. What we want to not is that, the bandwidth of the web is not unlimited and this procedures by all the users in a daily basis enlarges this problem.

Many well-known systems try to solve this problem by creating rss feeds or personalized micro-sites where a user can add his own interests and watch the most recent and popular issues on them. The rss feeds have become very popular and most of the news portals use them. But still, the problem is the filtering of information as the rss feeds are not intended for such a use. Regarding the personalization issue, the attempts that have been made from the major search engines and portals include only the issue of viewing already categorized content according to the user's interests. This means that the user is not included into the classification procedure.

MyYahoo! [12] is a very representative example as thousands of internet user visit it in a daily basis. After the login, the user is empowered with functionality that helps to personalize the page. More specifically, the user can add his special interests on news issues by selecting general topics from a list. Every time the user accesses the web page, the more recent results on the topic are displayed. This procedure seems very helpful but it does not include the user into the classification and rating procedure. Another representative example is the service that is provided by the Google and more specifically the news service [16]. The page that appears is fully customizable and the user can add his own query to the appearing results but his choice is not included in the categorization mechanism but only to the rating mechanism of the entire web.

In this paper, the proposed news portal architecture bases upon scalable text classification, in order to include the user in the classification procedure. Without having prior knowledge of user's interests, the system is able to provide him articles that match his profile. The user specifies the level of his expertise on different topics and the system relies on a new text analysis technique in order to achieve scalable classification results. Articles are decomposed into the vector representation of their sentences and classification bases upon the similarity of the category vectors and the sentences vectors (instead of the document-article vectors). This procedure enables the system to capture articles that refer to several topics, while their general meaning is different.

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. Subsequently, different solutions for this problem are described that base upon the reduction of the problem into multiple standard binary classification problems. Section 4 describes our Scalable Classification Algorithm that derives from spectral decomposition of the training documents into the vector representation of their sentences. In Section 5, the general architecture of a personalized news categorization system is presented and description of how personalization is implemented in order to exploit user's awareness of a topic and further enhance the categorization procedure is provided. Experimental evaluation both of the algorithm and the system is given in Section 5, using two different datasets (one widely used for standard text classification evaluation and one that consists of manually collected news from well-known web portals). Finally, Section 6 summarizes the results and introduces open issues.

II. 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 [22]:

Definition 1 (Standard Text Classification) Let $C = \{c_1 , \ldots, c_{|C|}\}$ 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. (R is the set of the real numbers)

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 is used for training of the mechanism which includes labeled documents that cannot be further expanded or limited. Moreover, definition of a specific category relies only on information that training documents provide. A classification function is specified by the minimization of an *effectiveness* measure [22] that shows how much functions $\breve{\Phi}$ and Φ 'coincide'. In tradition, this measure is based 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 from each other. However, we can argue that classification procedure still remains static, which means, 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, usually, come up with text classification problems. However, characteristics of these text classification problems involve dynamic changes of Web users' behavior and 'on-the-fly' definition of the category topics.

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

Consider, for example, the previous text article and Web users A and B. A is a journalist that needs 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.

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. Though, it is obvious that although user A would find useful 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, 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 the 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 $C=\{c_1 ,\dots,c_{|C|}\}$ a set of growing set of categories and $D=\{d_1,\dots,d_{|D|}\}$ a growing set of documents. A scalable text classifier is a function $\Phi = D\times C \rightarrow \Re^p$. (R is the set of Real Numbers)

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 specific m 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:

1. Defines p similarity classes and p corresponding similarity functions that map a document into a specific category c. Similarity classes can be shown as different ways to interpret the general meaning (concept) of a text document.

2. Permits the classification of each document into different categories depending on the similarity class that is used.

3. Permits the definition of new members and the erasure of existing ones from the categories set. This implies 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.

III. SOLUTIONS BASED ON STANDARD TEXT CLASSIFICATION TECHNIQUES

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) [23][17]. 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 afore-mentioned first technique that is the construction of many binary classifiers (one for each category).

A. *Scalable Naïve Bayes*

Naïve Bayes classifier treats a document d as a vector of k attributes $d = \{v_1, v_2, \ldots, v_k\}$. The naïve Bayes model assumes that all attribute values v_j, are independent given the category label c. Thus, a maximum a posteriori (MAP) classifier can be constructed as follows:

$$c^* = \arg\max_{c \in C} \left\{ P(c) \times \prod_{j=1}^{K} P(v_j|c) \right\} \qquad (1)$$

To cope with features that remain unobserved during training, the estimate of the $P(v_j | c)$ is usually adjusted by Laplace smoothing

$$P(v_j|c) \quad = \quad \frac{N_j^c + a_j}{N^c + a} \qquad (2)$$

where N_j^c is the frequency of attribute j in D^c, $N^c = \sum_j N_j^c$, and $a = \sum_j a_j$.

Introduction of different similarity classes can be done by modifying Equation 1 and change the decision about the category. Instead of choosing the category c that maximizes the a posteriori probability, we can just rank categories depending on that probability and then 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.

B. *Scalable Rocchio Classifier*

Rocchio is an early text classification method [20]. In this method, each document is represented as a vector, and each feature value in the vector is computed using the classic TD-IDF scheme [21]. Let D be the whole set of training documents, and C_j be the set of training documents in class c_j. Building a Rocchio classifier is achieved by constructing a prototype category vector $\vec{c}_j$ for each class c_j according to

Formula 3.

$$\vec{c}_j = \alpha \frac{1}{|C_j|} \sum_{\vec{d} \in C_j} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|D - C_j|} \sum_{\vec{d} \in D - C_j} \frac{\vec{d}}{\|\vec{d}\|}. \qquad (3)$$

α and β are parameters that adjust the relative impact of relevant and irrelevant training examples. [7] recommends α = 16 and β = 4. New documents are classified to the category that maximizes the cosine similarity measure [21].

Different similarity classes can be easily constructed by ranking categories according to the cosine similarity 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.

C. Scalable k nearest neighbors

k nearest neighbor classification (kNN) is a well-known statistical approach which has been intensively studied in pattern recognition for over four decades.

The kNN algorithm is quite simple: Given a test document, the system 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. Using a threshold criterion on these scores, binary category assignments are obtained.

The decision rule in kNN can be expressed as:

$$y(\vec{x}, c_j) = \sum_{\vec{d}_i \in kNN} sim(\vec{x}, \vec{d}_i) y(\vec{d}_i, c_j) - b_j \qquad (4)$$

where $y\left(\vec{d}_i, c_j\right) \in \{0,1\}$ is the classification for document d_i with respect to category c_j (y = 1 for YES, and y = 0 for NO); $sim\left(\vec{x}, \vec{d}_i\right)$ is the similarity between the test document $\vec{x}$ and the training document $\vec{d}_i$; and b_j is the category-specific threshold for the binary decisions. The category-specific threshold is automatically learned using a "validation set" of documents. 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.

IV. The Scalable classification algorithm

Given the above solutions to the standard classification problem, our approach should also specify the definition of the categories in a matter that would allow the algorithm to improve precision of future results. The algorithm shares a lot with the scalable Rocchio classifier and requires an initial set of predefined categories and their corresponding labeled data. As experimental analysis will prove, even if labeled data are noisy, the algorithm is able to identify abnormalities in some categories and determine how to further split the initial data into more categories.

A. Text analysis using sentences decomposition

We study decomposition of document vectors of the Rocchio classifier 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 infinite ways into a number of components.

THEOREM 1 Let $\vec{d}_i = [v_1, v_2, \dots, v_k]$ be the vector representation of document $\vec{d}_i$ and an integer $m > 0$. There exist at least two different decompositions of $\vec{d}_i$ into m different components.

PROOF For a v_i *write* $v_i = \sum_{j=1}^{m} a_j$ *and construct the* m *components as:*

$$\vec{d}_i = \sum_{j=1}^{m} \vec{b}_j \text{ where } \vec{b}_1 = [v_1, v_2, ..., a_1, \dots, v_k] \text{ and } \vec{b}_j = \left[0, 0, ..., a_j, \dots, 0\right],\ \forall j \le m$$

Given Theorem 1 it is easy to prove that:

THEOREM 2 Let $\vec{d}_i = [v_1, v_2, \dots, v_k]$ be the vector representation of document $\vec{d}_i$ and an integer $m > 0$. There exist non finite number of different decompositions of $\vec{d}_i$ into m different components.

PROOF We can apply recursively, k times Theorem 1 for one of the components of $\vec{d}_i$ resulting in k different decompositions. This stands for any k>0.

	D_1	D_2							···	D_n
			s_1	s_2	s_3	s_4	···	s_k		
t_1		a_1	a_{11}	a_{12}	a_{13}	a_{14}	···	a_{1k}		
t_2		a_2	a_{21}	a_{22}	a_{23}	a_{24}	···	a_{2k}		
t_3		a_3	a_{31}	a_{32}	a_{33}	a_{34}	···	a_{3k}		
t_4		a_4	a_{41}	a_{42}	a_{43}	a_{44}	···	a_{4k}		
t_5		a_5	a_{51}	a_{52}	a_{53}	a_{54}	···	a_{5k}		
t_6		a_6	a_{61}	a_{62}	a_{63}	a_{64}	···	a_{6k}		
t_7		a_7	a_{71}	a_{72}	a_{73}	a_{74}	···	a_{7k}		
t_8		a_8	a_{81}	a_{82}	a_{83}	a_{84}	···	a_{8k}		
t_9		a_9	a_{91}	a_{92}	a_{93}	a_{94}	···	a_{9k}		
···		···	···	···	···	···	···			
t_m		a_m	a_{m1}	a_{m2}	a_{m3}	a_{m4}	···	a_{mk}		

Figure 1: Example Term to Documents matrix, with term to sentences analysis of a specific document. Values aij satisfy equation: $a_i = \sum_{j=1}^{k} a_{ij}, \forall 1 \le i \le n$

Theorem 2 tells us, that whenever we use vector representation of a document we lose information. 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 $\vec{d}_i$. 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 $\vec{s}_k$ is a vector $\vec{s}_k = \left[v'_1, v'_2, \ldots, v'_{|s_k|}\right]$ representing k-th sentence of document d_i.

Using a decomposition that Definition 3 provides us, we can therefore compute the standard cosine similarity using Equation 4. A modified version of a 'term-to-document' matrix can also be used to include information about the sentences decomposition. Figure 1 provides an example

$$\cos\left(\vec{d}_i, \vec{c}_j\right) = \frac{\vec{d}_i \cdot \vec{c}_j}{\left\|\vec{d}\right\|\left\|\vec{c}_j\right\|} = \frac{\sum_{k=1}^{n} \vec{s}_k \cdot \vec{c}_j}{\left\|\sum_{k=1}^{n} \vec{s}_k\right\|\left\|\vec{c}_j\right\|} \qquad (4)$$

B. The Algorithm

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 a similarity threshold of the distance between a specific number of sentences of an article and some categories, in order to allow classification of the article in many categories. Formal definition of the Training Phase of the Scalable classification algorithm is shown in Algorithm 1:

Training Phase

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

Algorithm 1 Training Phase of the Scalable Classification Algorithm

Main characteristics of the classification phase (Algorithm 2) include (a) 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 (b) 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.

Classification Phase

1) Decompose unlabeled text document into its sentences
2) Compute term to sentences matrix of the document
3) Compute document vector by combining the columns of the term to sentences matrix
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
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)

7) The category vector computed during step 3 of Training Phase is re-computed based on the newly acquired data after the classification of the unlabeled text document to categories matching the threshold criterion

Algorithm 2: Classification Phase of the Scalable Classification Algorithm

V. PERSONALIZED NEWS CATEGORIZATION

Below, the architecture of a personalized news categorization system, that exploits scalability feature of the aforementioned algorithm so as to properly model personal profile.

A. General Architecture of the System

The system consists of distributed sub-systems that cooperate in order to provide end-user with categorized news articles from the web that meet his personal needs. The main features of the architecture are:

- Modularity: creation of autonomous subsystems

The core mechanism of the system we created can be described as a general manager and a main database. This is the module where everything starts from and concludes to. The subsystems of the mechanism can work autonomously but the general manager is responsible for the cooperation of them.

As we can see from Figure 2 the whole system consists of a manager, a database system and seven subsystems.

The crawler sub-system is responsible for fetching web documents that contain useful news articles. Except from a standard crawler mechanism, it also maintains a list of RSS urls from many major portals. Content extraction manager uses the web components technique [5], [6] and heuristics, in order to extract the text from the fetched web documents. Preprocessing manager, Keyword Extraction manager, Keyword – Document matcher and Dynamic Profile manager are implementing the Scalable Classification Algorithm that we introduced in the previous Section.

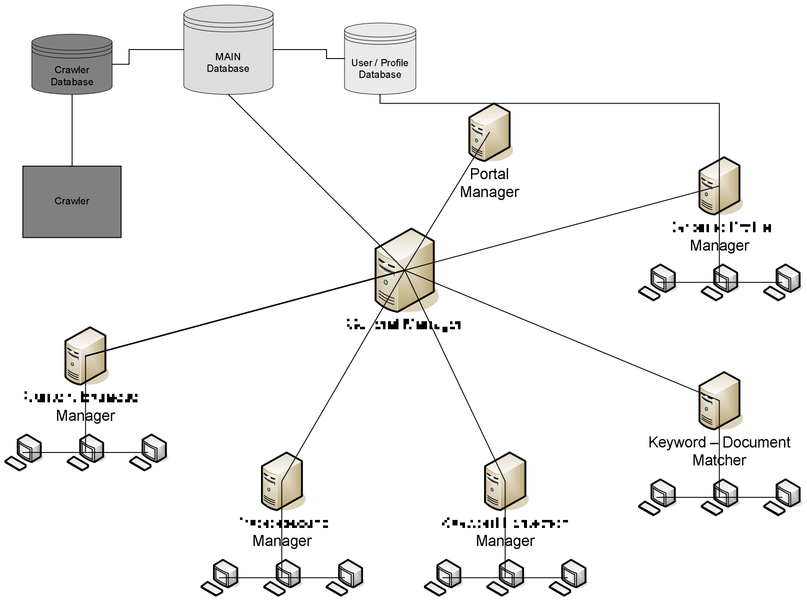


Figure 2. General Architecture

- Distributing the procedure:

The procedure of retrieving, analyzing and categorizing content from the World Wide Web is sequential because each step needs the previous to be completed in order to start. This does not preserve the implementation of a distributed system for the completion of each step, but introduces a limitation that step N+1 cannot be started if step N is not completed. This means that step N for the processing on text X can be completed in parallel with step N for the processing of text Y.

B. Scalability as Personalization

Users of the system select the level of their expertise on different categories. Using this information, the core mechanism of the system that implements the Scalable Classification Algorithm changes the number k of sentences (according to

Table 1) that should match the threshold criterion of a category in order to be classified.

k (number of sentences)	User expertise
1	low
2	medium
3	high

Table 1: Configuration of number of sentences that match the threshold criterion vs user expertise

VI. EXPERIMENTAL EVALUATION

Experimental evaluation involves two main steps. Firstly, we analyze the performance of the Scalable Classification algorithm, using two well known datasets. Using data gathered during this procedure, we also specify different criterion thresholds and apply them to the core mechanism of the presented system. At last, experimental results of the real articles' classification are presented.

A. Tuning the Scalable Classification Algorithm

In order to evaluate our scalable classification technique we used the 20 newsgroup dataset [8], that is a widely used dataset in the evaluation process of many classification algorithms (both supervised and unsupervised). However, we also developed a new dataset that consists of news articles collected from well-known web portals.

1) 20 newsgroup dataset

The 20 newsgroup dataset is a collection of articles of 20 newsgroups. Each category contains 1000 articles. We preprocessed the documents so as to use only the main text (as Subject section may contain many keywords of the corresponding category). In order to evaluate the similarity values between different category vectors we used the standard metric [13] that computes the cosine of the corresponding vectors a_j and q using Formula 2.

$$\cos\theta_j = \frac{a_j^T q}{\| a_j \|_2 \| q \|_2} = \frac{\sum_{i=1}^{t} a_{ij} q_i}{\sqrt{\sum_{i=1}^{t} a_{ij}^2}\sqrt{\sum_{i=1}^{t} q_i^2}} \qquad \textbf{(2)}$$

Angles between category vectors of this dataset can be seen in Table 1.

2) News dataset[1]

This dataset consists of five general categories (business, education, entertainment, health, and politics) and includes RSS articles from different well- known web portals (bbc.co.uk, cnn.com, usatoday.com).

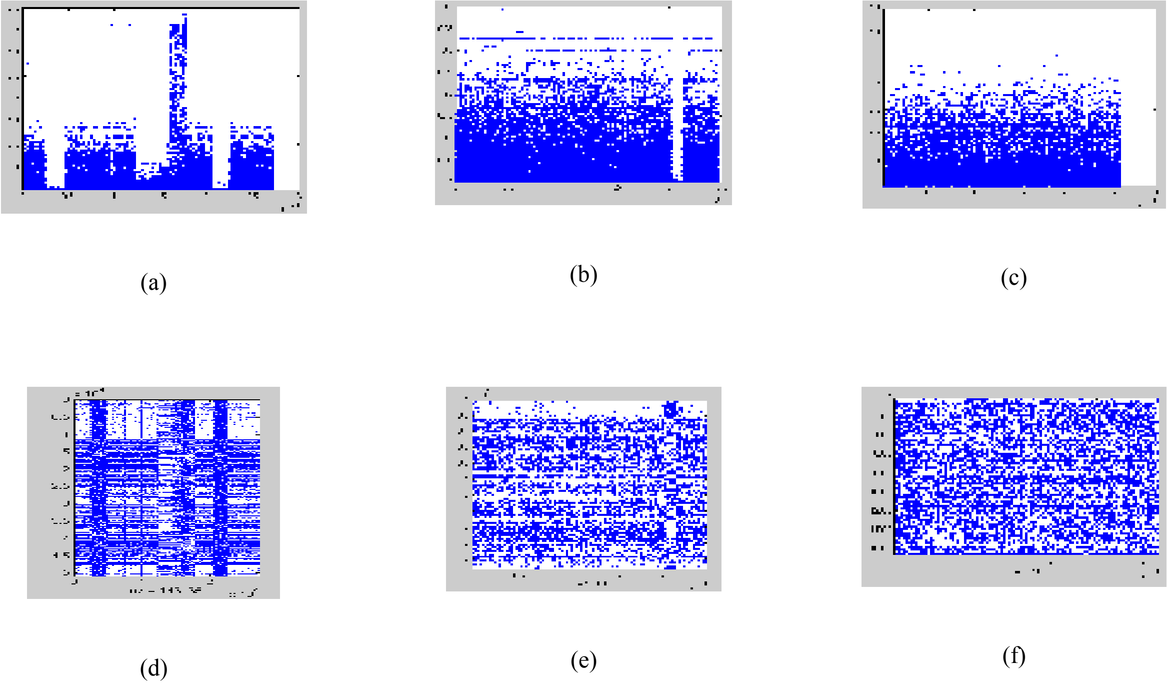
Category		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1	0	34,4	42,7	32,5	31,7	36,5	32,4	27,5	28,6	27,2	32,5	33,4	26,2	27,5	34,2	41,3	26,0	37,9	28,0	19,4
comp.graphics	2	34,4	0	38,5	29,0	28,9	28,0	33,8	33,4	34,6	33,2	36,4	35,3	29,0	32,4	36,4	48,4	35,0	44,0	38,3	37,4
comp.os.ms-windows.misc	3	42,7	38,5	0	37,5	38,9	32,2	40,1	41,7	42,7	41,6	44,3	43,9	39,2	41,8	45,3	53,8	42,8	50,2	44,0	43,4
comp.sys.ibm.pc.h	4	32,5	29,0	37,5	0	21,1	29,9	30,7	29,9	30,3	30,7	34,7	33,9	25,9	30,9	36,3	47,9	32,3	43,1	36,2	35,0
comp.sys.mac.hard	5	31,7	28,9	38,9	21,1	0	32,0	29,3	29,9	32,0	29,3	33,1	35,3	25,4	30,2	35,2	46,1	32,0	42,1	36,4	34,8
comp.windows.x	6	36,5	28,0	32,2	29,9	32,0	0	36,2	34,8	34,4	35,6	38,5	35,4	31,3	34,8	39,0	50,5	36,6	45,5	39,7	38,8
misc.forsale	7	32,4	33,8	40,1	30,7	29,3	36,2	0	29,1	32,4	29,3	32,6	37,5	26,3	30,4	35,5	46,2	31,9	42,2	33,1	32,4
rec.autos	8	27,5	33,4	41,7	29,9	29,9	34,8	29,1	0	21,6	23,5	29,1	33,5	22,7	26,3	33,4	45,6	27,0	39,9	31,9	30,9
rec.motorcycles	9	28,6	34,6	42,7	30,3	32,0	34,4	32,4	21,6	0	25,8	31,0	32,5	24,9	27,7	34,9	47,7	28,3	41,3	32,9	31,5
rec.sport.baseball	10	27,2	33,2	41,6	30,7	29,3	35,6	29,3	23,5	25,8	0	22,9	35,2	23,7	26,3	33,1	43,6	27,8	39,3	32,8	31,8
rec.sport.hockey	11	32,5	36,4	44,3	34,7	33,1	38,5	32,6	29,1	31,0	22,9	0	39,6	28,2	31,2	36,4	46,6	32,7	42,1	36,8	35,9
sci.crypt	12	33,4	35,3	43,9	33,9	35,3	35,4	37,5	33,5	32,5	35,2	39,6	0	29,3	30,6	35,9	50,7	32,2	43,1	33,9	34,2
sci.electronics	13	26,2	29,0	39,2	25,9	25,4	31,3	26,3	22,7	24,9	23,7	28,2	29,3	0	19,9	27,1	43,7	25,9	38,1	30,8	29,5
sci.med	14	27,5	32,4	41,8	30,9	30,2	34,8	30,4	26,3	27,7	26,3	31,2	30,6	19,9	0	27,9	44,6	27,7	39,5	31,3	30,4
sci.space	15	34,2	36,4	45,3	36,3	35,2	39,0	35,5	33,4	34,9	33,1	36,4	35,9	27,1	27,9	0	48,3	34,6	43,3	37,0	36,6
soc.religion.christi	16	41,3	48,4	53,8	47,9	46,1	50,5	46,2	45,6	47,7	43,6	46,6	50,7	43,7	44,6	48,3	0	45,3	49,0	46,8	43,0
talk.politics.guns	17	26,0	35,0	42,8	32,3	32,0	36,6	31,9	27,0	28,3	27,8	32,7	32,2	25,9	27,7	34,6	45,3	0	33,9	21,6	24,4
talk.politics.mideas	18	37,9	44,0	50,2	43,1	42,1	45,5	42,2	39,9	41,3	39,3	42,1	43,1	38,1	39,5	43,3	49,0	33,9	0	32,7	36,7
talk.politics.misc	19	28,0	38,3	44,0	36,2	36,4	39,7	33,1	31,9	32,9	32,8	36,8	33,9	30,8	31,3	37,0	46,8	21,6	32,7	0	21,6
talk.religion.misc	20	19,4	37,4	43,4	35,0	34,8	38,8	32,4	30,9	31,5	31,8	35,9	34,2	29,5	30,4	36,6	43,0	24,4	36,7	21,6	0

Table 2: Angles between category vectors of the 20-newsgroup dataset

[1] The news dataset version used in this paper is publicly available at http://students.ceid.upatras.gr/~antonell/news_dataset.tar.gz

3) Results

We present evaluation of the similarity thresholds obtained for the ‘sentence vs. category’ using the 20 newsgroup dataset as well as an overview of the classification accuracy of the algorithm based on its ability to identify abnormalities in the definition of a category and automatically decide on further splitting a category into many, so as to sustain uniform ‘sentence vs. category’ similarity distributions. All experiments were conducted using data collected using both the Rainbow tool [19] for statistical analysis and separation procedures of the datasets, as well as using the TMG [25] a recently developed MATLAB toolbox for the construction of term document matrices from text collections.



(a)

(b)

(c)

(d)

(e)

(f)

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 (second line) (a) comp.os.ms-windows.misc (b) comp.windows.x (c) talk.politics.misc

Table 2 presents the sentence vs. category vectors similarities for different categories of the 20 newsgroup dataset. The basic results can be summarized as:

- General categories (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 and proceed on further separation, 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 in the previous Section are shown to have a matrix with blocked structure (e.g. (d) or (e) matrices).

	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

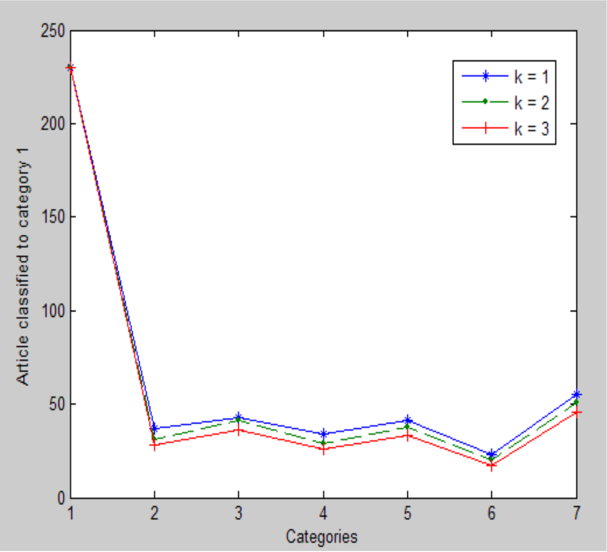
Table 3: Average F-scores for different number of similarity classes

Evaluation of the accuracy of the algorithm can be seen on Table 3, where average F-scores are presented for different numbers of total similarity classes.

Experimental setup included the creation of a test set of documents that were constructed as the result of joining p documents from different categories into one document. Precision and recall were computed according to the number of these categories that the algorithm computed.

A. System Evaluation

Using the similarity threshold of 19.43 degrees that we computed using the 20 newsgroup dataset, we tuned the core mechanism of the system that uses the Scalable Classification Algorithm so as to classify an article into a category if k sentences of the article match this criterion. Figure 4 shows how many business articles are also classified to other categories for three values of k. As value of k increases, the amount of multi-labelled articles decreases.



	Category
1	Business
2	Education
3	Entertainment
4	Health
5	Politics
6	Sports
7	Technology

Figure 4: Multi-labeled business articles for different values of k (number of sentences to match the threshold criterion)

We also, tested the classification feedback that our Scalable Classification Algorithm provides. Figure 5, reports the maximum and the minimum angle between the different category vectors, as time passes and newly classified articles affect the category vectors. We run the system for a period of 15 days and we computed the angles between the re-computed category vectors at the end of every day. It is easily seen that minimum angles vary close to 20 degrees, while maximum angles are close to 40 degrees.

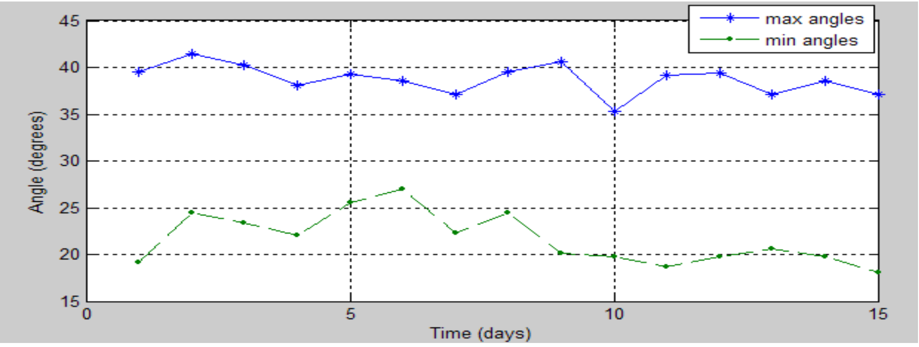


Figure 5: Maximum and Minimum angles between category vectors, for a period of 15 days. Classification feedback of our algorithm results in small variances of the vectors that represent each category.

VII. Future Work

Future work will include further exploration of the presented text analysis technique and direct use of it for other web mining problems. There is also need for development of well-specified datasets for the evaluation of future algorithms on the scalable classification problem. Finally, it will be interesting to further explore the classification of real articles using our system and apply data mining techniques on data deriving from the amount of multi-labeled documents, trying to identify the behavior and impact of major ‘alarm news’.

VIII. Conclusions and Discussion

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 tha 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.

As an application of the combination of the text analysis technique and the scalable classification algorithm, we propose a new approach to personalized news categorization that exploits user’s awareness of a topic in order to classify articles in a ‘per-user’ manner. Furthermore, the architecture of the backend of a portal that uses this technique is presented and analyzed. Unlike standard techniques for personalization, user only specifies his level of expertise on different categories.

IX. ACKNOWLEDGEMENTS

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

[1] D. Achlioptas, F. McSherry, Fast Computation of Low Rank Matrix Approximations, STOC '01 ACM.

[2] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia, Data mining through spectral analysis, STOC '01 ACM.

[3] M.W. Berry, S.T. Dumais & G.W. O' Brien, Using Linear Algebra for Intelligent Information Retrieval, UT-CS-94-270, Technical Report.

[4] M. W. Berry, Z. Drmac, E. R. Jessup, Matrices, Vector Spaces, and Information Retrieval, SIAM Review Vol. 41, No 2 pp 335-362.

[5] C. Bouras, V. Kapoulas, I. Misedakis, A Web - page fragmentation technique for personalized browsing, 19th ACM Symposium on Applied Computing - Track on Internet Data Management, Nicosia, Cyprus, March 14 - 17 2004, pp. 1146 – 1147.

[6] C. Bouras and A. Konidaris, Web Components: A Concept for Improving Personalization and Reducing User Perceived Latency on the World Wide Web, Proceedings of the 2nd International Conference on Internet Computing (IC2001), Las Vegas, Nevada, USA, June 2001, Vol. 2, pp.238-244.

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

[8] CMU Text Learning Group Data Archives, 20 newsgroup dataset, http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html.

[9] P. Drineas, R. Kannan, A. Frieze, S. Vempala, V. Vinay, Clustering of large graphs via the singular value decomposition, Machine Learning 56 (2004), 9-33.

[10] P. Drineas, R. Kannan, M. Mahoney, Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix, Tech.Report TR-1270, Yale University, Department of Computer Science, February 2004.

[11] S. Dumais, G. Furnas, T. Landauer, Indexing by Semantic Analysis, SIAM.

[12] Google News Service, http://news.google.com

[13] 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.

[14] G. Kumaran and J. Allan, “Text classification and named entities for new event detection”, *SIGIR '04*

[15] T. K. Landauer, P. W. Foltz, D. Laham (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, pp. 259-284.

[16] My Yahoo!, http://my.yahoo.com

[17] 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

[18] C. H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala, Latent semantic indexing: A probabilistic analysis, 17th Annual Symposium on Principles of Database Systems (Seattle, WA, 1998), 1998, PP 159-168.

[19] Rainbow, statistical text classifier, http://www-2.cs.cmu.edu/~mccallum/bow/rainbow/.

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

[21] Salton, G. and McGill, M. (1983). “Introduction to Modern Information Retrieval”. McGraw-Hill.

[22] F. Sebastiani, “Machine Learning in automated text categorization”, *ACM Comput. Surv 2002.*, Vol 34, No 1, pp 1 – 47

[23] Y. Yang, “An evaluation of statistical approaches to text categorization”, *Journal of Information Retrieval*, Vol 1, No. 1/2, pp 67--88, 1999

[24] Y. Yang and X. Liu, “A re-examination of text categorization methods“, *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, (SIGIR), 1999, pp 42-49

[25] 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).