Evolutionary Feature Selection for Text Documents using the SVM

Daniel I. Morariu, Lucian N. Vintan, and Volker Tresp

Abstract—Text categorization is the problem of classifying text documents into a set of predefined classes. After a preprocessing step, the documents are typically represented as large sparse vectors. When training classifiers on large collections of documents, both the time and memory restrictions can be quite prohibitive. This justifies the application of feature selection methods to reduce the dimensionality of the document-representation vector. In this paper, we present three feature selection methods: Information Gain, Support Vector Machine feature selection called (SVM_FS) and Genetic Algorithm with SVM (called GA_SVM). We show that the best results were obtained with GA_SVM method for a relatively small dimension of the feature vector.

Keywords—Feature Selection, Learning with Kernels, Support Vector Machine, Genetic Algorithm, and Classification.

I. INTRODUCTION

WHILE more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of document content. Document categorization is one solution to this problem. In recent years a growing number of categorization methods and machine learning techniques have been developed and were applied in different contexts.

Documents are typically represented as vectors in a features space. Each word in the vocabulary is represented as a separate dimension. The number of occurrences of a word in a document represents the value of the corresponding component in the document's vector. This document representation results in a huge dimensionality of the feature space, which poses a major problem to text categorization. Due to the large dimensionality, much time and memory are needed for training a classifier on a large collection of documents. For this reason we explore various methods to reduce the feature space and thus the response time. As we'll show the categorization results are better when we work with a smaller optimized dimension of the feature space. As the feature space grows, the accuracy of the classifier doesn't grow significantly; actually it even can decrease due to noisy

D. Morariu is with the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Computer Science Department, E. Cioran Street, No. 4, 550025 Sibiu, Romania (phone: 40/0740/092202; e-mail: daniel.morariu@ulbsibiu.ro).

L. Vintan is with the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Computer Science Department, E. Cioran Street, No. 4, 550025 Sibiu, Romania (e-mail: lucian.vintan@ulbsibiu.ro).

V. Tresp is with the Siemens AG, Information and Communications, 81739 Munchen, Germany (e-mail: volker.tresp@siemens.com).

vector elements [6].

This paper present a new method for feature selection that uses Genetic Algorithm with a fitness function based on the Support Vector Machine method. We have also studied the influence of the input data representation on classification accuracy. We have used three types of representation, Binary, Nominal and Cornell Smart. For the classification process we used the Support Vector Machine technique, which has proven to be efficient for nonlinearly separable input data [8], [9].

The Support Vector Machine (SVM) is actually based on learning with kernels and support vectors. A great advantage of this technique is that it can use large input data and feature sets. Thus, it is easy to test the influence of the number of features on classification accuracy. We implemented SVM classification for two types of kernels: "*polynomial kernel*" and "*Gaussian kernel*" *(Radial Basis Function - RBF)*. We employed a simplified form of the kernels using more intuitive parameters. We have also modified this SVM representation so that it can be used as a method of features selection (SVM_FS and GA_SVM).

Section 2 and 3 contain prerequisites for the work that we present in this paper. In section 4 we present the framework and the methodology used for our experiments. Section 5 presents the main results of our experiments. The last section debates and concludes on the most important obtained results and proposes some further work.

II. SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) is a classification technique based on statistical learning theory [8], [11] that was applied with great success in many challenging non-linear classification problems and on large data sets.

The SVM algorithm finds a hyperplane that optimally splits the training set. The optimal hyperplane can be distinguished by the maximum margin of separation between all training points and the hyperplane. Looking at a two-dimensional problem we actually want to find a line that "best" separates points in the positive class from points in the negative class. The hyperplane is characterized by a decision function like:

$$f(x) = \text{sgn}(\langle \mathbf{w}, \mathbf{\Phi}(x) \rangle + b) \tag{1}$$

where $\mathbf{w}$ is the weight vector, orthogonal to the hyperplane, "b" is a scalar that represents the margin of the hyperplane, "x" is the current sample tested, "$\Phi(x)$" is a function that transforms the input data into a higher dimensional feature space and $\langle \cdot, \cdot \rangle$ representing the dot product. *Sgn* is the sign

function. If **w** has unit length, then <**w**, Φ*(x)*> is the length of Φ*(x)* along the direction of **w**. Generally **w** will be scaled by ‖**w**‖. The training part the algorithm needs to find the normal vector "**w**" that leads to the largest "*b*" of the hyperplane.

For extending the SVM algorithm from two-class classification to multi-class classification typically one of two methods is used: "One versus the rest", where each topic is separated from the remaining topics, and "one versus the one", where a separate classifier is trained for each class pair. We selected the first method for two reasons: First, preliminary experiments shows that the first method gives better performance, which might be explained by the fact that the Reuter's database contains strongly overlapping classes and assigns almost all samples in more than one class. Second overall training time is much shorter for the first method.

III. FEATURE SELECTION METHODS

A substantial fraction of the available information is stored in text or document databases which consist of a large collection of documents from various sources such as news articles, research papers, books, web pages, etc. Data stored in text format is considered semi-structured data that means neither completely unstructured nor completely structured. In text categorization, feature selection is typically performed by assigning a score or a weight to each term and keeping some number of terms with the highest scores while discarding the rest. After this, experiments evaluate the effects that feature selection has on both the classification performance and the response time.

Numerous feature scoring measures have been proposed and evaluated: Odds Ratio [4], Information Gain, Mutual Information [4], Document Frequency, Term Strength [1], or Support Vector Machine [5], a. o.

A. Information Gain

Information Gain and Entropy [4] are functions of the probability distribution that underlie the process of communication. The entropy is a measure of uncertainty of a random variable. Based on entropy, as attribute effectiveness, a measure is defined for features selection, called "Information Gain", and is the expected reduction in Entropy caused by partitioning the samples according to this attribute. The Information Gain of an attribute relative to a collection of samples S, is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

where *Values(A)* is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A is equal to v. Forman in [2] reported that Information Gain failed to produce good results on an industrial text classification problem, as Reuter's database. The author attributed this to the property of many feature scoring methods to ignore or to remove features needed to discriminate difficult classes.

B. SVM Feature Selection (SVM_FS)

Mladenic et Al. [5], present a method for selecting features based on a linear Support Vector Machine. The authors compare more traditional feature selection methods, such as Odds Ratio and Information Gain, in achieving the desired tradeoff between the vector sparseness and the classification performance. The results indicate that for the same level of sparseness, feature selection based on normal SVM yields better classification performances. In [3] the advantages of using the same methods in the features selection step and in the learning step are explained.

Following this idea we have used the SVM algorithm, with linear kernel, for feature selection. Thus the feature selection step becomes a learning step that trains using all features and calculates the (optimal) hyperplane that splits best the positive and negative samples. We obtain for each topic from the initial set the specified weight vector (the weight vector has the input space dimension) using linear kernel (multi-class classification). In contrast to Mladenic, we normalized all weight vectors, obtained for each topic. We make an average over all weight vectors and obtain the weight vector used in the subsequent step. Using this weight vector we select only the features with a weight having an absolute value greater then a specified threshold.

C. Genetic Algorithm for Feature Selection (GASVM)

Genetic algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply genetic operators to these structures so as to preserve critical information [12, 13, 14]. In our feature selection problem the chromosome is considered to be of the following form:

$$c = (w_1, w_2, ..., w_n, b) \quad (3)$$

where w_i, $i = \overline{1, n}$ represent the weight for each feature, and *b* represent the bias of the hyperplane of SVM. We consider that the training set has the form $\{\langle \vec{x}_i, y_i \rangle, i = 1, ..., m\}$, where y_i represents the output for the input sample $\vec{x}_i$, and it can only take -1 and +1. We chose this form of chromosome to facilitate using of SVM for fitness function, keeping into the chromosome the parameters that are modified into SVM decision function (1). Thus potential solutions to the problem encode the parameters of the separating hyperplane, *w* and *b*. In the end of the algorithm, the best candidate from all generations gives the optimal values for separating hyperplane orientation *w* and location *b*. Following the idea proposed for multi-class classification ("one versus the rest"), we try to find the best chromosome for each of the 24 considered Reuters topics. For each topic we start with a generation of 100 chromosomes, each of them having values randomly generated between -1 and 1.

Using the SVM algorithm with linear kernel $\langle w, x \rangle + b$ we can compute the fitness function for each chromosome. The evaluation through the fitness function is defined as:

$$f(c) = f((w_1, w_2, ..., w_n, b)) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b), \quad (4)$$

where $\mathbf{x}$ represents the current sample and n represents the number of features. In the next step we generate the next population using selection, crossover or mutation [12, 15].

The evolutionary process stops after a predefined number of steps are taken or when in the last 20 steps no change occurs.

At the end of the algorithm, we obtain for each topic the best chromosome that represents the decision function. We then normalize each weight vector in order to obtain all weights between 0 and 1. For selecting the best features we make an average over all those 24 obtained weight vectors and select the features according to their descending weights.

The general scheme of the genetic algorithm is presented in pseudo code in the Algorithm 1:

```
Begin
 For each topic from a topics_set
  begin
   generate a population
   while not terminated condition
    For each chromosome from population
     compute the fitness functions
    make next population:
    • select parents
    • recombine pairs of parents
    • apply mutation to offspring
   End while.
  • Store the chromosome that split the
    best the training set
  End for.
 • Take all stored chromosomes
 • Select features that have the
   greater absolute value.
End.
```
Algorithm 1: Pseudocode for GA feature selection algorithm

IV. EXPERIMENTAL FRAMEWORK

A. The Dataset

Our experiments are performed on the Reuters-2000 collection [10], which has 984Mb of newspapers articles in a compressed format. Collection includes a total of 806,791 documents, with news stories published by Reuters Press covering the period from 20.07.1996 through 19.07.1997. The articles have 9822391 paragraphs and contain 11522874 sentences and 310033 distinct root words. Documents are pre-classified according to 3 categories: by the *Region* (366 regions) the article refers to, by *Industry Codes* (870 industry codes) and by *Topics* proposed by Reuters (126 topics, 23 of them contain no articles). Due to the huge dimensionality of the database we will present here results obtained using a subset of data. From all documents we selected the documents for which the industry code value is equal to "System

software". We obtained 7083 files that are represented using 19038 features and 68 topics. We represent documents as vectors of words, applying a stop-word filter (from a standard set of 510 stop-words) and extracting the word stem. From these 68 topics we have eliminated those topics that are poorly or excessively represented. Thus we eliminated those topics that contain less than 1% documents from all 7083 documents in the entire set. We also eliminated topics that contain more than 99% samples from the entire set, as being excessively represented. After doing so we obtained 24 different topics and 7053 documents that were split randomly in training set (4702 samples) and evaluation set (2351 samples). In the feature extraction part we take into consideration both the article and the title of the article.

B. Kernel Types

The idea of the kernel is to compute the norm of the difference between two vectors in a higher dimensional space without representing those vectors in the new space. In practice we can see that by adding a constant bias to the kernel involves better classifying results. In this work we present results using a new idea to correlate this bias with the dimension of the space where the data will be represented. More information about this idea can be found in our previous work [7]. We consider that those two parameters (the degree and the bias) need to be correlated in order to improve the classification accuracy.

We'll present the results for different kernels and for different parameters for each kernel. For the polynomial kernel we vary the degree and for the Gaussian kernel we change the parameter C according to the following formulas (x and x' being the input vectors):

• Polynomial

$$k(x, x') = \left(2 \cdot d + \langle x \cdot x' \rangle\right)^d \quad (5)$$

• d being the only parameter to be modified

• Gaussian (radial basis function RBF)

$$k(x, x') = \exp\left(-\|x - x'\|^2 / n \cdot C\right) \quad (6)$$

• C being the classical parameter and n being the new parameter, introduced by us, representing the number of elements from the input vectors that are greater than 0.

As linear kernel we used the polynomial kernel with degree 1. For feature selection with SVM method and fitness function from GA we used only the linear kernel.

C. Correlating Parameters for the Kernel

Usually when learning with a polynomial kernel researchers use a kernel that can be expressed as $\left(\langle \mathbf{x} \cdot \mathbf{x'} \rangle + b\right)^d$ where d and b are independent parameters. Parameter "d" is the kernel degree and it is used as a parameter that helps mapping the input data into a higher dimensional space. Thus, this parameter is intuitive. The second parameter "b" (the bias), is not so easy to infer. In all studied articles, the researchers used a nonzero b, but they didn't present a method for selection it. We notice that if this parameter was eliminated (i.e., chosen to

be zero) the quality of the results can be poor. It is logically that there is a need to correlate the parameters d and b because the offset b needs to be modified as the dimension of the space modifies. Due to this, based on running laborious classification simulations presented in [7], we suggest the best correlation is "$b=2*d$".

Also for the Gaussian kernel we modified the standard kernel used in the research community given by formula $k(x,x') = \exp(-\|x - x'\|^2 / C)$, where the parameter C is a number witch usually takes values between 1 and total numbers of features. We introduce the parameter n [7] that multiplies the usually parameter C with a value that represents the number of distinct features having weights greater than 0 that occur in the current two input vectors, decreasing substantially the value of C (see Equation 6). As far as we know, we are the first authors proposing a correlation between these two parameters for both polynomial and Gaussian kernels.

D. Representing the Input Data

Because there are many ways to define the feature-weight, we represent the input data in three different formats, and we try to analyze their influence on the classification accuracy. In the following formulas $n(d, t)$ is the number of times that term t occurs in document d, and $n(d, \tau)$ is the maximum frequency occurring in document d.

- **Binary representation** – in the input vector we store "0" if the word doesn't occur in the document and "1" if it occurs.
- **Nominal representation** – we compute the value of the weight using the formula:

$$TF(d,t) = \frac{n(d,t)}{\max_{\tau} n(d,\tau)} \tag{7}$$

- **Cornell SMART representation** –we compute the value of the weight using the formula:

$$TF(d,t) = \begin{cases} 0 & if\ n(d,t) = 0 \\ 1 + \log(1 + \log(n(d,t))) & otherwise \end{cases} \tag{8}$$

This are later called as BIN, NOM or CS.

V. EXPERIMENTAL RESULTS

A. Feature Selection for Multi-Class Classification

For a fair comparison between the three proposed feature selection methods, we need to use the same number of features. For the Information Gain method the threshold for selecting the features represents a value between 0 and 1. For the other two methods the features number is the same with the number of features obtained through Information Gain method.

In what follows we present the influence of the number of features regarding to the classification accuracy for each input data representation and for each feature selection method, considering 24 distinct classes. We present results only for a numbers of features smaller or equal to 8000. In [6] we

showed that for a number of features greater than 8000, the classification accuracy doesn't increase, sometimes even decreases.

The classification performance, as it can be observed from Fig. 1 and Fig. 2, is not improved when the number of features increases. We notice that there is a slight increase in the accuracy when we raise the percentage of features from the initial set from 2.5% (475 features) to 7% (1309 features). The accuracy doesn't increase for a larger percentage of selected features. More than this, if more than 42% of the features were selected, the accuracy slightly decreases [6, 16]. This can occur because the additional features can be noisy.

The SVM algorithm depends of the order of selecting input vectors, finding different optimal hyperplanes when the input data are selected in different order. Genetic algorithm with SVM fitness function stipulates this in feature selection step. The SVM_FS and GA_SVM obtained comparable results but there are better comparatively with Information Gain.

In Fig. 1 the influence of number of features in classification accuracy obtained for all feature selection presented methods are presented. In the classification step we use the SVM algorithm with polynomial kernel and Cornell Smart data representation. In Fig. 2 are presented results obtained using Gaussian kernel and binary data representation. As can be observed (average bar in fig. 2) GA_SVM obtain better results with Gaussian kernel comparatively with others two methods. So, GA_SVM is better in average with 1% comparatively with SVM_FS (84.27% for GA_SVM comparatively with 83.19% for SVM_FS) and with 1.7% comparatively with IG (84.27% for GA_SVM and 82.58% for IG). For polynomial kernels SVM_FS obtain in average better results with 0.9% comparatively with IG (from 86.24% for SVM_FS to 85.31% for IG) and with 0.8% comparatively with GA_SVM (from 86.24% to 85.40%). In almost all cases the best results are obtained for a small numbers of features (in average for 1309).

The training time for polynomial kernel with degree 2 and SVM_FS method increases from 11.52 seconds for 475 features to 14.56 seconds for 1306 features and to 46.55 seconds for 2488 features. Thus for fast learning we need a small numbers of features and as it can be observed from Fig. 1, with SVM_FS method we can obtain better results with a small number of features Also the time needed for training using features selected with IG or GA_SVM are usually greater than the time needed for training with features selected using SVM_FS (for example, for 1309 features it takes 14.56 seconds for SVM-FS versus 26.42 seconds for IG and 18.14 seconds for GA_SVM).

For Gaussian kernel the time is on average (for all made testes) with 20 minutes greater then the time needed for training the polynomial kernel for both features selected with IG or SVM_FS. The numbers are given for a Pentium IV at 3.4 GHz, with 1GB DRAM and 512KB cache, and WinXP.
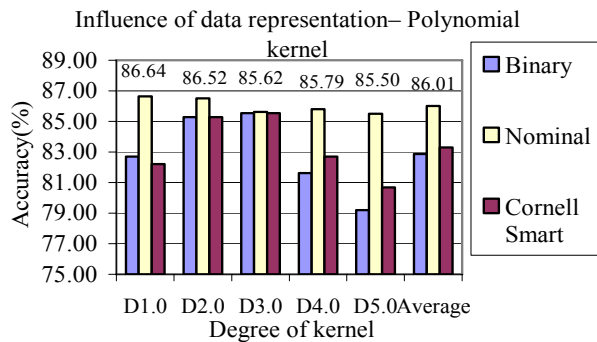
Fig. 1 Influence of data representation and degree of the kernel for polynomial kernel
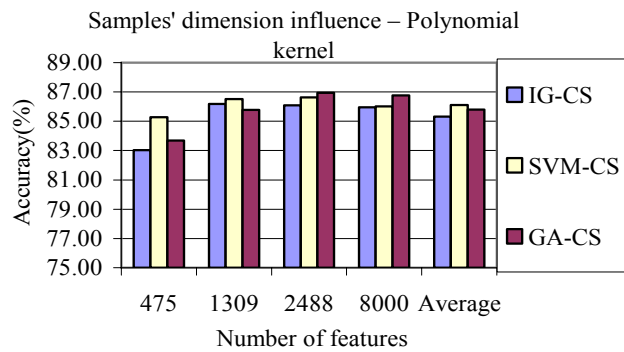

Fig. 3 Influence of the number of features on the classification accuracy using Polynomial kernel with degree equal to 2
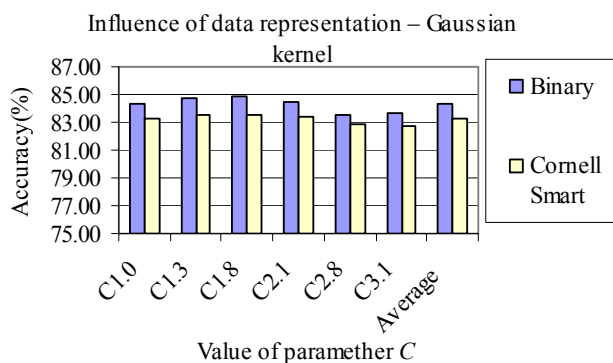

Fig. 2 Influence of data representation and parameter *C* for Gaussian kernel
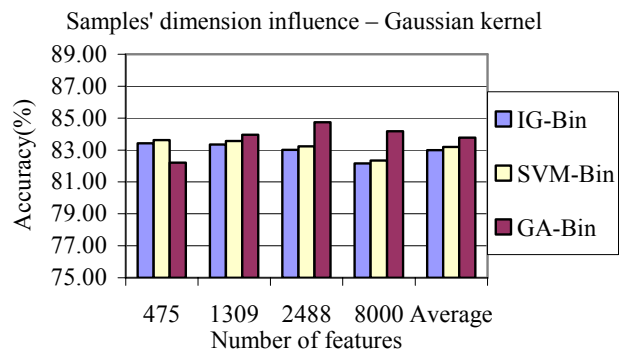

Fig. 4 Influence of the number of features on the classification accuracy using Gaussian kernel with parameter C equal to 1.3

B. Influence of Kernel Degree and Data Representation

In order to find a good combination of kernel type, kernel degree and data representation we run eleven tests: five tests for a polynomial kernel (Fig. 3), and respectively six tests for a Gaussian kernel with different values for the parameter C (Fig. 4). In [6, 16] we report additional results. In Fig. 3 we present results obtained for polynomial kernel and SVM_FS feature selection method with a data set with 1309 features, which proved to be the best number (see Fig. 1, 2).

Fig. 3 shows that text files are generally linearly separable in the input space (if the input space has the right dimensionality) and the best results were obtained for a small kernel degree (1 and 2). Taking into consideration data representation for all five tests, best results were obtained with nominal representation that obtained in average 86.01% in comparison with binary representation (82.86%) or Cornell Smart (83.28%).

In Fig. 4 we present results obtained for Gaussian kernel for two types of data representation and for five distinct value of parameter C, using a data set with 1309 features obtained with GA_SVM method. We chose this method to present results here because it obtained best results in first tests (see Fig. 2). Into Gaussian kernel (Fig. 4) we add a parameter that

represents the number of elements greater then zero (parameter "*n*" from equation 6). Nominal representation (equation 7) represents all weight values between 0 and 1. When parameter "*n*" is used, all the weights become very close to zero involving very poor classification accuracies (for example, due to its almost zero weight, a certain word really belonging to the document, might be considered to not belong to that document). So we don't present here the results obtained using the nominal representation.

For 1309 feature space dimension, the GA_SVM method achieved an average accuracy of 83.91% in comparison with the SVM feature selection that achieved an average accuracy of 84.90% and for Information Gain was achieved 84.62%). In Table I we present all average accuracies obtained and we can observe that the SVM_FS method obtains better results for each dimension of the data set for polynomial kernel. Also we can observe that the average accuracy doesn't increase so much when the dimension of the set increases (especially for SVM_FS). The SVM_FS method obtains best results with a small dimension of the features space (85.03% for 475 features) in comparison with IG that needs more features (8000 features for 84.72%) for obtaining its the best results. Genetic algorithm feature selection method needs also 8000

features for obtain best results 85.01%.

In Table II we compute the average over all tested values for the Gaussian kernel. For Gaussian kernel GA_SVM feature selection method the results are better comparatively with SVM_FS, and in both case the results are greater than results obtained with IG (see Table I and Table II, IG columns). In comparison with results obtained using the polynomial kernel the results obtained using Gaussian kernel are smaller, whatever of feature selection method used.

TABLE I. AVERAGE ACHIEVED OVER ALL DATA SETS TESTED FROM POLYNOMIAL KERNEL AND NOMINAL REPRESENTATION

Method Nr. Features	IG	SVM_FS	GA_FS
475	82.91	**85.03**	81.94
1309	84.62	**84.90**	83.91
2488	84.54	**85.02**	84.90
8000	84.72	82.42	**85.01**

TABLE II. AVERAGE ACHIEVED OVER ALL DATA SET TESTED FOR GAUSSIAN KERNEL

Method Nr. Features	IG	SVM_FS	GA_FS
475	83.27	**83.31**	81.93
1309	83.33	83.39	**83.41**
2488	83.07	83.02	**84.02**
8000	82.20	82.42	**83.32**

In all presented results when we used SVM technique (in SVM_FS, GA_SVM and classification step) we used kernels presented into equations 5 and 6 with correlating parameters (*d* and *b* for polynomial kernel and *n* with input vectors for Gaussian kernel). In [7] we showed that this method assures better results in almost all cases.

VI. CONCLUSIONS AND FURTHER WORK

In this paper, we investigated whether feature selection methods can improve the accuracy of document classification. Three types of feature selection methods were tested and three types of input data representations were used. The best results were obtained when we chose a small (but relevant) dimension of the data set. After selecting relevant features, we showed that using between 2.5% to 7% from the total number of features, the classification accuracies are significantly better (with a maximum of 86.64% for SVM_FS method, polynomial kernel and nominal data representation). If we further increase the number of features to more than 10%, the accuracy does not improve or even decreases (to 86.52% for 2488 and 85.36 for 8000 features). When we used SVM_FS, better classification accuracy is obtained using a small number of features (accuracy of 85.28% for 475 features, representing about 3% from the total number of features) - needing small training time. Generally speaking, the SVM_FS method and GA_SVM were better than IG and both obtained comparable results. We have also observed that the polynomial kernel

obtains better results when we used a nominal data representation and the Gaussian kernel obtains better results when we used Cornell Smart data representation. The best accuracy was obtained by the Polynomial kernel with a degree of one, nominal representation and SVM_FS (86.64%) in comparison with Gaussian kernel that obtained only 84.85% accuracy for C=1.3, Cornell Smart representation and GA_SVM but for a greater numbers of features (2488). Also we showed that the training classification time increases only by 3 minutes, as the number of features increases from 485 to 1309 and increases by 32 minutes when number of features increases from 1309 to 2488.

Work is ongoing to classify larger text data sets (the complete Reuters database). In this work we want to develop a pre-classification of all documents, obtaining fewer samples (using simple algorithms like Linear Vector Quantization or Self Organizing Maps). After that we'll use the obtained samples as entry vectors for the already developed features selection and classification methods.

REFERENCES

[1] S. Chakrabarti, "Mining the Web- Discovering Knowledge from hypertext data", Morgan Kaufmann Press, 2003.
[2] G. Forman, "A Pitfall and Solution in Multi-Class Feature Selection for Text Classification", Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
[3] T. Jebara, "Multi Task Feature and Kernel Selection for SVMs", Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
[4] T. Mitchell, "Machine Learning", McGraw Hill Publishers, 1997.
[5] D. Mladenic, J. Brank, M. Grobelnik and N. Milic-Frayling, "Feature Selection Using Support Vector Machines", The 27th Annual International ACM SIGIR Conference (SIGIR2004), pp 234-241, 2004.
[6] D. Morariu, "Classification and Clustering using Support Vector Machine", 2nd PhD Report, University „Lucian Blaga" of Sibiu, September, 2005, http://webspace.ulbsibiu.ro/ daniel.morariu/html/Docs /Report2.pdf.
[7] D. Morariu, L. Vintan, "A Better Correlation of the SVM kernel's Parameters", Proceeding of The 5th RoEduNet International Conference, Sibiu, June 2006.
[8] C. Nello, J. Swawe-Taylor, "An introduction to Support Vector Machines", Cambridge University Press, 2000.
[9] J. Platt, "Fast training of support vector machines using sequential minimal optimization". In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods – Support Vector Learning, pages 185-208, Cambridge, MA, 1999, MIT Press.
[10] Reuters Corpus: http://about.reuters.com/researchandstandards/corpus/. Released in November 2000.
[11] B. Schoelkopf, A. Smola, "Learning with Kernels, Support Vector Machines", MIT Press, London, 2002.
[12] Whitely, D., A genetic Algorithm Tutorial, Foundations of Genetic Algorithms, ed. Morgan Kaufmann
[13] G, F. Luger, W. A. Stubblefield, *Artificial Intelligence*, Addison Wesley Longman, Third Edition, 1998
[14] G. Kim, S. Kim, *Feature Selection Using Genetic Algorithms for Handwritten Character Recognition*, Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, Amsterdam, 2000

[15] A. E. Eiben, J. E. Smith, *Introduction to evolutionary computing*, Springer-Verlag, 2003

[16] D. Morariu, L. Vintan, V. Tresp, *Feature Selection Methods for an Improved SVM Classifier*, Proceedings of the 14th International Conference on Computational and Information Science, pp 83-89, Prague, August 2006