Self-Organizing Word Map for Context-Based Document Classification

N. Tsimboukakis and G. Tambouratzis
Institute of Language and Speech Processing,
Artemidos 6 & Epidavrou Str., Amaroussion,
Athens, 151 25, Greece
email: {ntsimb, giorg_t }@ilsp.gr

Keywords: document classification, word map, hybrid neural network architecture, SOM, MLP

Abstract — In this paper, a novel SOM-based system for document organization is presented. The purpose of the system is the classification of a document collection in terms of document content. The system possesses a two-level hybrid connectionist architecture that comprises (i) an automatically created word map using a SOM, which functions as a feature extraction module and (ii) a supervised MLP-based classifier, which provides the final classification result. The experiments, which have been performed on Modern Greek text documents, indicate that the proposed system separates effectively the different types of text.

1 Introduction

The amount of documents that are available in modern databases (both physical and virtual) and can be accessed directly over the internet increases at an expanding rate. The organization and indexing of these documents are extremely important in order to render the information contained in them accessible. The exploitation of this information for the benefit of the citizen of the digital society depends directly on the ease and precision of the retrieval process.

The organization of massive document collections is a time-consuming task when performed manually. On the other hand, systems that automatically organize documents in terms of their content can prove cost–effective, provided that they can generate a systematic organization which is flexible, accurate and appears intuitive to the users. To that end, the system presented here is a small step as it aims to assign documents to predefined categories, which coincide with their content. The experiments performed prove the effectiveness of the system.

The motivation for the proposed system came from earlier document organization systems based on the SOM model which have been developed for the English language (for instance [1],[2]). Such systems have the advantages of performing an autonomous clustering of documents, based on the similarity of texts in terms of features. Though the SOM has been studied extensively for document processing applications, it is mainly the choice of features that differentiates the applications. In most cases the features chosen are frequencies of occurrence of words or terms within the documents ([1],[2],[3]). Though the SOM is in general one of the most scaleable neural network models, when large document collections are processed it is impossible to take into account the frequencies of all words in the documents. Consequently, several approaches have been proposed to reduce the number of features, the most common one being the random projection of the frequency matrix to lower dimensions [4]. This approach is used in the WEBSOM [1],[2] and the Marginal median SOM (MM-SOM) [5].

Word frequencies, but at a sentence level, have been proposed by Pullwitt [6], in order to enhance the SOM-based text-document clustering using a two-stage architecture, where both stages consist of SOM models.

Alternative features have also been evaluated as inputs for the SOM model. For instance, Linden [7] has evaluated linguistic types of information such as part-of-speech and syntactic level features with respect to their effectiveness in self-organized document maps. In a related study, the effectiveness of different features on SOM-based systems discriminating the authors of documents has been investigated [8].

Georgakis et al. [5] use a variation of the SOM word map as a feature extractor. The word map is created by measuring for each word context–related vectors of high dimensionality. Random projection is used to reduce the dimensionality of feature vectors [4]. At a second stage, MM-SOM is used to cluster documents.

In the current article, the document collections processed are in the Greek language. The use of texts from a highly-inflectional language results in a number of word forms substantially higher than in languages such as English. Therefore, the texts need to be processed in order for the words contained to be transformed into a lemmatized form. Another issue that is addressed in the present work is the systematic selection of word-frequency features, making use of the information content of each word. Finally, a three-stage hybrid architecture is proposed, where the two neural network layers employ different neural models to perform the classification of documents.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

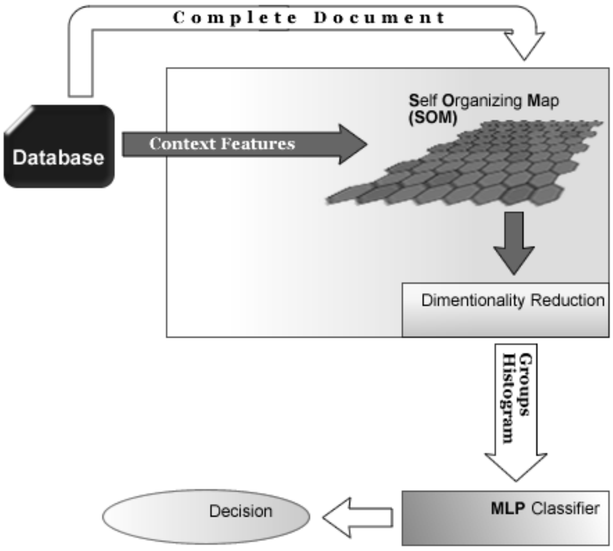
It is claimed that the combination of two different networks provides a better performance, the first-layer, unsupervised network achieving a dimensionality-reduction, while the second-layer, supervised network provides a more accurate clustering.

2 System Description

The document organization system contains three basic modules (Figure 1):

- The word map module, where words are grouped in terms of their concept using a SOM structure. This module serves to create in an unsupervised manner groups of words, each group possessing similar characteristics.
- The dimensionality reduction module, where word map groups are further combined to create fewer, but larger word groups. Each neuron of the SOM lattice can be viewed as one group, though to further reduce the number of groups, similar nodes (and thus the words assigned to these nodes) can be concatenated into larger groups.
- The classifier module, where a supervised classifier is used to assign documents to categories. This is performed by making use of the groups of words generated from the previous two modules. Within the present article, a neural network architecture is used to perform the classification, this being an MLP network.

Figure 1: System description



Within the remainder of this section, the document organization method is presented in detail. Though the proposed method has been developed and thereafter evaluated within a variety of document collections, it is illustrated here with the aid of a specific example, in order to demonstrate the intermediate results of each stage. The system presented here has been developed using object-oriented techniques and the C++ programming language.

2.1 Dataset

The dataset used for illustrating the behaviour of the proposed system comprises texts from the proceedings of the Greek Parliament. These Minutes contain a large amount of texts spanning almost two centuries, edited via a well-established procedure by specialised personnel and available in electronic format. From the Minutes, a set of documents have been collected which comprise a total of 5 speakers for a given Parliamentary period (a 4-year period between two consecutive general elections). The full dataset contains 1,004 documents, which have a document length varying from just over 100 up to 8,000 words (more details on the dataset are provided in [9]). These documents have been independently assigned by specialized linguists to categories on the basis of their content [10], as shown in Table 1. Out of these texts, a total of 560 documents have been assigned to one of the four most popular categories, namely internal affairs, external affairs, education, and economy.

The Greek language possesses an extremely rich morphology. Due to this fact, a word may appear in several different forms, greatly increasing the number of distinct words appearing in documents. This can cause serious problems if the different word forms originating from the same stem are not projected onto the corresponding stem itself. To overcome this bottleneck, the documents are pre-processed and converted to lemmatized form using the ILSP tagger-lemmatizer [11].

Table 1: Distribution of documents in the data set

Category	Number of documents
Internal Affairs	82
External Affairs	207
Education	70
Economy	201
Remaining categories	444
Total	**1004**

2.2 Word Map

The organization of documents into context-based categories requires the extraction of measurable quantitative features. In earlier SOM applications such as the WEBSOM [1][2], mostly lemma frequencies were used as discriminating features. Due to the large number of lemmas available within documents of a general (unconstrained) language, the dimensionality of the feature vector for each new document increases. However, the large dimensionality leads to excessive requirements for processing power and eventually to an intractable implementation. In order to avoid this bottleneck, a subset of lemmas is selected either with SVD or a random projection method [4].

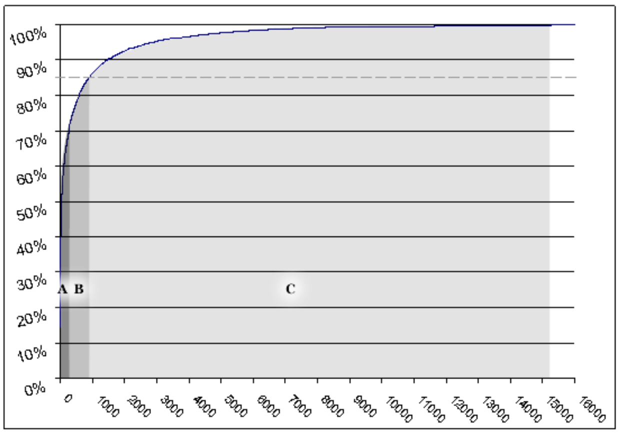
In the proposed system, lemmas are first organized into groups on a Self Organizing Map. The dimensionality of the feature vector used as input to the SOM is significantly lower than the number of lemmas available in the dataset. For each lemma, the ideal case would be to record the number of co-appearances with all lemmas in the dataset. Such a solution would highly increase the dimensionality of the lemma feature vectors. To avoid the increase in dimensionality, only a subset of available lemmas was used as the feature set, so that every lemma could be described by its co-occurrences with the feature set lemmas.

The selection of feature lemmas was made by an ad-hoc rule. The contribution or frequency of appearance for each lemma in the whole corpus is assumed to follow Zipf's distribution. For example there are only a few lemmas that are extremely frequent and many lemmas that are very rare. As an analogy to Zipf's law, Pareto's principle (also known as the 80-20 rule) states that 20% of causes are responsible for the 80% of the results. Pareto's principle, also termed the ABC analysis, has been mostly applied to quality control and management tasks. According to the ABC analysis a portion of the causes is characterized as A which indicates very important events, with B and C corresponding to less important and to unimportant events respectively. In this case category A contains highly-frequent lemmas (which correspond to functional words, such as articles, auxiliary verbs and conjunctions), B contains frequently-used lemmas and C contains rare lemmas. It seems appropriate to select the feature lemmas from category B, since lemmas within this frequency range do not correspond to functional or very common words (which bear no meaning that reflects a specialized content) and yet are frequent enough to describe the remaining lemmas. The limits of the ABC analysis are set as follows (the relative sizes of the three categories being shown in Figure 2):

- category A contains the most frequent lemmas that collectively amount to 70% of all appearances (257 lemmas are included in category A).
- category B contains lemmas that contribute the next 15% (70-85%) of all appearances (634 lemmas are included in category B).
- category C contains lemmas that correspond to the remaining 15% of appearances (16,197 lemmas are placed in category C).

However, lemmas that appear less than three times are omitted from the remainder of the analysis, as their frequency of occurrence is too low to provide measurable frequencies, and thus category C is left with 6,987 lemmas. The total of lemmas retained in the analysis is thus equal to 7,244 (6,987+257 lemmas). The context frequencies of these lemmas (categories A and C) with respect to the 634 feature lemmas of Category B are to be used to cluster the documents.

Figure 2: ABC classification of lemma based on the frequency of appearances



More specifically, each of the 7,244 lemmas is represented by a vector of 634 elements, where each element corresponds to one B-lemma and indicates the number of times the given lemma occurs in the same sentence as that B-lemma within the document dataset. It must be noted that only sentence-level appearances are counted and if a lemma or a feature appears more than once within a sentence it only counts for one. The use of sentences as a basis for counting the appearances of lemmas is based on the principle that a full-stop between sentences is the least ambiguous point at which the description of an idea or event is terminated, while in subsequent sentences completely unrelated concepts may be presented. The components for each vector of word coappearances are defined in the following equation:

$$s_f(w_i) = \frac{N(w_i, f)}{N(w_i)} \qquad (1)$$

where $s_f(w_i)$ is the measurement for feature f of lemma w_i, $N(w_i, f)$ is the number of sentences where both lemma w_i and feature f appear and $N(w_i)$ is the number of sentences where lemma w_i appears. Each vector is normalized so that its components sum to one. The normalized version of the previous formula is:

$$\hat{s}_f(w_i) = \frac{s_f(w_i)}{\sum\limits_{forall\_f} s_f(w_i)} \qquad (2)$$

where $\hat{s}_f(w_i)$ is the normalized representation.

It must be noted here that, though the approach of Georgakis et al. [5] also creates word maps, it differs substantially in the method used to select features and count word vectors, as the order of appearance of the words in the sentence is taken into account. This approach considers a narrow window of one word before and after the word to be described. The approach presented here is better suited to the Modern Greek language, where there are no strict syntactic word-order rules (each word can be placed almost anywhere in a sentence without altering its meaning). In the system presented here only word co-occurrences in sentences are counted, which gives lower-dimensional feature vectors (here the number of features is *M*, i.e. equal to the number of B-lemmas, while in Georgakis [5] the number of features is *3N-2*, where *N* is the number of words-stems considered). Additionally Georgakis et al. [5] employ a random projection method [4], while in the proposed system a feature selection procedure is preferred.

Since each lemma can be described by a numeric vector, it is straightforward to use a common clustering algorithm such as the Self-Organizing Map to group lemmas that appear in similar contexts, and thus possess similar vectors. It is reasonable to assume that words which appear in similar contexts bear related meanings. SOM models have been reportedly applied to word grouping tasks in the past [12].

In brief, SOM networks usually comprise a single-layer structure of neurons, where each neuron is connected to every input. Each of the neurons contains a weight vector with dimensionality equal to the input dimensionality. The SOM model operates in a competitive manner, as the output of the training process is the identity of the neuron that best matches the pattern presented, in terms of the predefined distance. This winning neuron is then adapted towards the training pattern. The neurons are connected with their direct neighbours depending on the lattice geometry selected. The most commonly used lattice is the 2-dimensional hexagonal structure as among the possible geometries it approximates more closely a uniform distribution of neurons over the lattice. The hexagonal structure is used in the experiments described here.

The SOM training process is unsupervised and thus desired outputs for input patterns are not provided. The neuron weights are initialised linearly along the two eigenvectors of the training data autocorrelation matrix that correspond to the two greatest eigenvalues. After the initialisation, every pattern is presented to the network and the winner neuron is adapted so as to increase its similarity to the pattern. All neighbouring lattice neurons are also updated to a decreasing degree as their distances on the lattice from the winner increase. The function used to update neurons is usually a Gaussian kernel function of the distance on the lattice. The weight update formula is:

$$\Delta \vec{W}_i = \eta h_{i,c(X)} (\vec{X} - \vec{W}_i) \qquad (3)$$

where ΔW_i is the weight change for the i^{th} neuron, η is the learning rate, *h* is the neighbourhood function and *c(x)* is the winner neuron for training pattern *X*.

In order to stabilise the learning procedure, the batch training algorithm suggests that weight updates take place only after the entire training set is presented. Then each neuron weight update equals the average of the data vectors within its neighbourhood:

$$\vec{W}_i = \frac{\sum_j h_{i,c(x_j)} \cdot \vec{X}_j}{\sum_j h_{i,c(x_j)}} \qquad (4)$$

The neighbourhood radius is decreased during training, so that the SOM network finally converges to a stable state. For the experiments reported here, an environment in C++ was developed, this being a port of the original SOM toolbox [13].

The size of the lattice used was automatically determined from the data using the estimation procedure provided by the SOM toolbox. In the experiments reported here the map has a size of 22x19 neurons. Lemmas that were assigned to category B of the ABC analysis were omitted from the map, since they are present in the feature vectors. The B-lemmas give very high values for the corresponding feature (since $s_f(w_i = f) = 1$ from equation 1) and they could influence heavily the training algorithm, substantially more than the other lemmas. The distribution of A and C lemmas on the SOM lattice is presented in figure 3.

It should be noted that different variants of the map generation method have been evaluated, depending on whether the context neighbourhood extends either over an entire sentence or over a limited radius in terms of words. It has been found that the use of a radius limit results in a less uniform distribution of words over the map. On the contrary, when the context is defined over the entire sentence, the number of words over each node is more similar (for instance, in the case of figure 3, only three nodes are assigned with more than 60 words and no unused nodes exist). When the context measurements within any sentence are limited by a weighted function of the form $exp(\frac{1-x}{L})$, where x is the distance between a lemma and a lemma feature and L a radius parameter, the number of SOM nodes with more than 60 words assigned is equal to five (this indicating a less uniform clustering of words on the SOM map as shown in figure 4) while there are sixteen nodes corresponding to empty groups. In order to estimate in an objective manner the number of cluster centers in the SOM map, the Davies-Bouldin index was used. This resulted in an index value of 20.68 and 20.57 for the maps of figures 3 and 4 respectively, indicating the existence of 20 cluster centres in both cases.

Different lattice sizes for the SOM model were also tested but in the absence of a reference clustering, results cannot be clearly evaluated. Additionally, assigning words into groups can yield completely different results even when performed manually by different experts. The evaluation of the system effectiveness shall be addressed in section 3.

Figure 3: Distribution of lemmas from categories A and C on the SOM lattice, with each node annotated with the number of lemmas assigned to that node

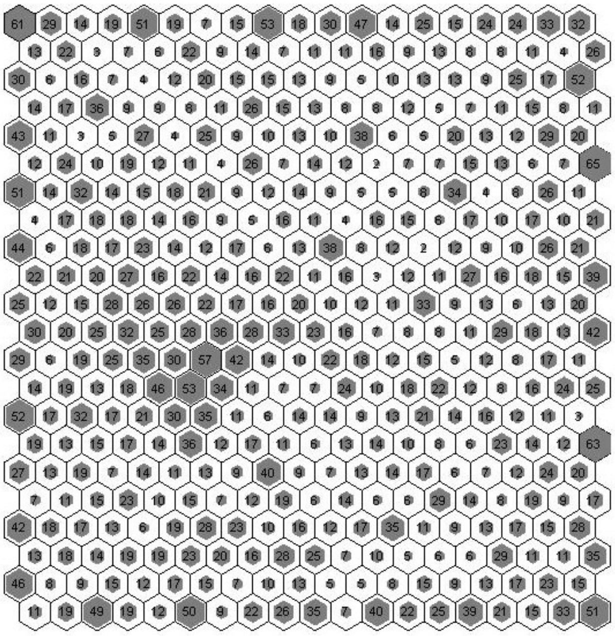
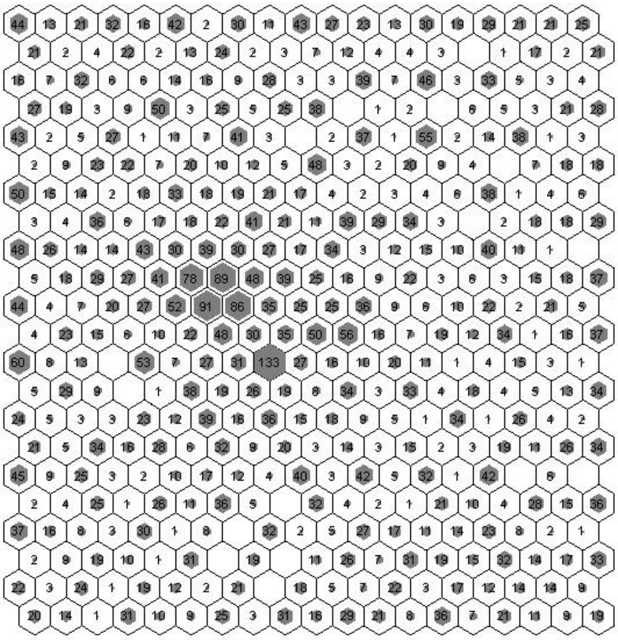


Figure 4: Distribution of lemmas on the SOM lattice when using weighted measurements



2.3 Dimensionality Reduction

When using a SOM map to cluster data, the number of nodes is most frequently chosen to be higher than the expected number of clusters. This is due to the fact that the clustering method is unsupervised and thus it is likely that neighboring nodes correspond to the same actual class. Since the number of groups that are input to the next stage of the system (the classifier stage) influences highly the classifier complexity, further compression of the map would be desirable. For instance, the document dataset used in the present experiments includes 560 labeled documents from the four most common categories and an input dimensionality of 418(=22x19 nodes) for a multilayer perceptron of three layers would be high. A three layer MLP with four outputs contains a total of $(418+1)h+(h+1)4$ parameters (biases and weights), where h is the number of neurons in the hidden layer.

To further cluster the word map, a batch k-means algorithm was used (similarly to [14]). An alternative for this stage could involve a second-level SOM. However, the use of a SOM rather than a k-means algorithm would increase the computational complexity of the task. Furthermore, the requirement to reach a clustering result with k clusters coupled with the topology-preserving characteristics of a SOM model would necessitate the use of a larger SOM network at this second level, and a subsequent clustering phase to reach a total of exactly k clusters.

The batch k-means [15] algorithm is initialized evenly with respect to the 2-D word map, so that the distance on the grid between two neighboring centers at each dimension is almost constant. Initial k-means centers are placed on the vertexes of a parallelogram lattice, where each of the two edges of the parallelogram lattice has a length proportional to the SOM lattice size. For example, when choosing k =75 and the dimensionality of the SOM lattice is 22x19, the initial k-means centers lattice will contain $\left[\sqrt{75\frac{22}{19}}\right]=\left[9.32\right]=9$ centers across the first dimension and $\left[\sqrt{75\frac{19}{22}}\right]=\left[8.05\right]=8$ across the second. Note that the actual number of cluster centers is 72 ($9\cdot 8$) rather than 75.

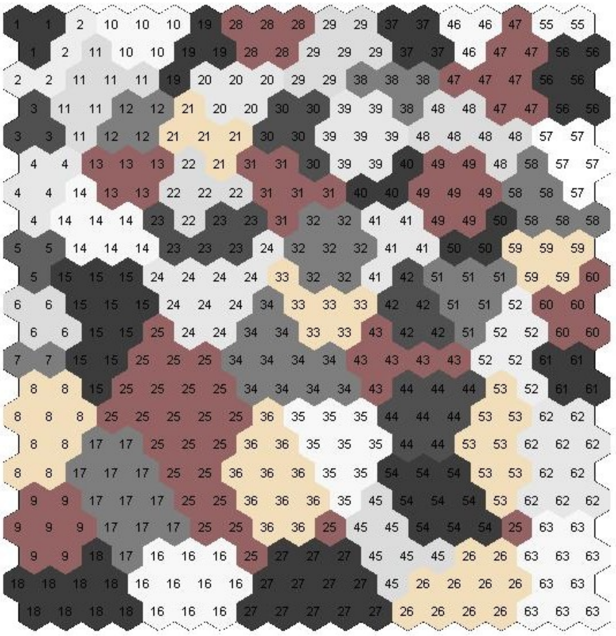
Each initial center is given the value of the nearest neuron's vector, and thus has a dimensionality of 634. The SOM grid is actually taken into account only at the initialization step, while the distance between centers is calculated from their vectors. Each iteration of the algorithm consists of two distinct phases [15]. Firstly, every one of the 418 codebooks is assigned to the nearest k-means center. Secondly, each center is updated to the mean of the vectors that were assigned to it:

$$\vec{c}_i(t+1)=\frac{1}{N}\sum_{j=1}^{N}\vec{d}_j \qquad (5)$$

Where $\vec{c}_i(t+1)$ is the center i at iteration $t+1$, N is the number of patterns that were assigned to $\vec{c}_i(t)$ and $\vec{d}_j$ is the j^{th} pattern assigned to $\vec{c}_i(t)$.

Following the dimensionality reduction process, there are k groups of lemmas. The value of k could describe the level of complexity and detail that the user requires from the clustering system. A typical example of the k-means result is shown in figure 5, where the SOM nodes of figure 3 have been colored to indicate the resulting groups of nodes (and thus the corresponding groups of words). As can be seen, the resulting groups consist mainly of neighboring nodes, the only exception involving group 25, which comprises two disconnected nodes in addition to the main group of 23 connected nodes. This is a consequence of the topographical error of SOM due to the dimensionality reduction from the 634-dimensional input space to the 2-dimensional lattice.

Figure 5: Grouping of the 418 nodes of the SOM map in terms of similarity.



2.4 Supervised Neural Classifier

The grouping of lemmas generated by the k-means algorithm module is subsequently used as a feature extraction module to describe the documents of the dataset. For each document, the number of lemmas that belong to each group is counted. This set of measurements comprises the representative vector of a document, which has a dimensionality of k. Each vector is normalized so that the sum of its components is equal to one.

The classifier algorithm chosen is a Multilayer Perceptron (MLP) trained with the Resilient RPROP backpropagation variation [16][17].–The MLP consists of layers of neurons, where each neuron is stimulated by an activation signal, which is a weighted sum of the outputs of previous-layer neurons. It has been reported [18] that MLPs are capable of creating any input (features) to output (decision) mapping, by using a single hidden layer that contains an adequate number of neurons. Every neuron processes its activation signal through a non-linear bipolar activation function (such as the hyperbolic tangent function) and transmits the output signal to the next-layer neurons through synaptic weights.

The most widely used method for supervised training of an MLP is the family of backpropagation algorithms. In backpropagation, the deviation of the network's output from a desired target is measured by a mean square error function. Backpropagation is based on determining the error in the network at the output layer and then propagating the error signal from the output towards the input layer, one layer at a time. In the present study, the RPROP variant of backpropagation is used since, in comparison to other variants tested it resulted in a faster convergence, higher accuracy and greater consistency over a set of test runs for the given task.

The RPROP algorithm [16][17] adapts independently weights in predetermined steps, which are different for each weight and do not depend on the error function gradient. Each weight is increased when its error function derivative (error signal) is negative; otherwise it is decreased by a step value, as determined by (6):

$$\Delta w_{ij}(t+1)=\begin{cases} -\,g_{ij}(t), & \dfrac{dE(t)}{dw_{ij}}>0 \\ +\,g_{ij}(t), & \dfrac{dE(t)}{dw_{ij}}<0 \\ 0, & \dfrac{dE(t)}{dw_{ij}}=0 \end{cases} \qquad (6)$$

where Δw_{ij} is the change of the weight connecting the i^{th} node of the current layer to the j^{th} neuron of the next layer and g_{ij} is the update step of weight w_{ij}. When the error function derivative (gradient) with respect to the weight retains the same sign for a consecutive number of training steps, the step value (g_{ij}) is increased by a fraction to make the network converge faster. On the other hand, when consecutive sign changes are observed, the weight step value is decreased to make the procedure more accurate near local minima, as indicated by (7):

$$g_{ij}(t+1)=\begin{cases} n^{+}\cdot g_{ij}(t), & \dfrac{dE(t)}{dw_{ij}}\cdot\dfrac{dE(t-1)}{dw_{ij}}>0 \\ n^{-}\cdot g_{ij}(t), & \dfrac{dE(t)}{dw_{ij}}\cdot\dfrac{dE(t-1)}{dw_{ij}}<0 \\ 0, & \dfrac{dE(t)}{dw_{ij}}\cdot\dfrac{dE(t-1)}{dw_{ij}}=0 \end{cases} \qquad (7)$$

where n^{+} is a predefined constant step increase value (with $n^{+}>1$) and n^{-} is a step decrease value (with $0<n^{-}<1$).

The chosen MLP architecture comprises three layers. The input layer contains k neurons, where k is equal to the number of word groups generated by the k-means algorithm. The output layer contains a number of neurons equal to the document categories used. In order to improve the training procedure, the Nguyen – Windrow initialization technique is adopted [19]. This technique ensures that weights are initially uniformly distributed in the input space.

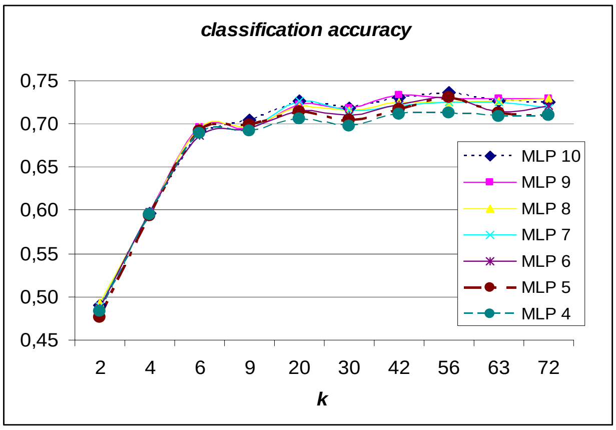
3 Experiments

In order to determine the effectiveness of the proposed system in the task of context-based document classification, various system setups were tested. The main parameters of the classification system proposed here are the value of k and the size of the hidden layer of the MLP. Since a deterministic procedure is not available for defining the number of hidden-layer neurons, this has been determined by experiments with network sizes ranging from 4 to 10 hidden neurons.

The value of k was varied between 76 and 2 groups. Due to the k-means initialization procedure, where the distance between neighbor centers should remain constant at each dimension (as reported in section 2.3) it is not possible to use every value of k but an integer value near k as explained in section 2.3. So the values tested here were: 72, 63, 56, 42, 30, 20, 10, 6, 4, and 2.

For the 560 documents of the dataset that belong to the four topic categories, nine subsets were created. Seven of these were used to form the training set, one subset as validation set for the early stopping of the MLP training and the final subset as the test set for evaluating the system performance. All possible combinations of sets were run and the average performance on the test set is reported.

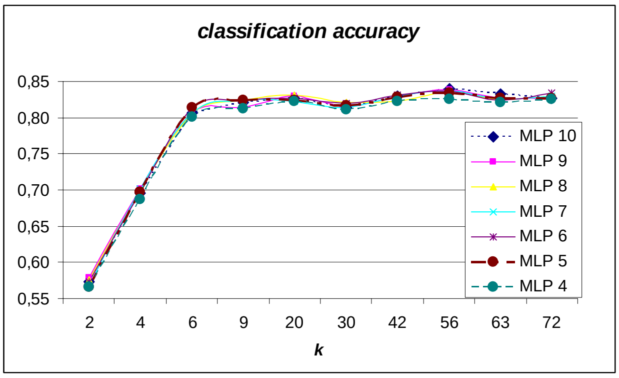
Figure 6: Classification accuracy for the four category discrimination task.



The classification accuracy observed for the four-topic discrimination task is almost 70% for most experiments. The maximum accuracy observed was 74% for k =56 and an MLP with 10 hidden neurons. It can be seen in figure 6 that varying the MLP size affects the performance by less than 2%. The most important parameter, as expected, is the value of k. Small k values lead to low classification accuracy while k values larger than 6 give satisfactory results. Finally it can be seen that by increasing k to values larger than 56 leads to a slightly reduced performance.

Due to the fact that topic classification can be ambiguous in terms of each individual's background and preferences, a second experiment was also carried out. From the topic categories, the most general category ("Internal Affairs") was removed. Then, the same experiments were run for the three-category classification task. The results presented in figure 7 show that the classification accuracy increased by almost 10% in comparison to the 4-topic task. The best accuracy observed is 84% for k =56 and an MLP with 10 hidden neurons. Once again the MLP size does not affect system performance considerably. Performance is quite acceptable for k values larger than 6, as it exceeds 80%. Additionally for large k values ($k > 56$), the performance deteriorates slightly (by approximately 1%) with respect to the maximum value.

Figure 7: Classification accuracy for the three category discrimination task.



4 Conclusions

Within this article, a method has been proposed for the organization of a document collection in terms of content. This method uses three distinct stages (corresponding to distinct modules), of which only the last one is supervised, in order to result in the desired classification. The first two stages are performed in the absence of any human guidance and are intended to gradually generate a grouping of words into contextually similar (and, it is expected, semantically-related groups). Input regarding the category identity is limited to the third and final stage, where the final classification of documents is performed.

The method presented here possesses a number of similarities to SOM-based systems [1][5], which have already been proposed for document clustering using the SOM model as a fundamental module and choosing as features the frequencies of occurrence of specific lemmas. However, the present approach proposes certain different solutions, for instance systematically selecting specific features on the basis of their relative frequency of occurrence in the texts. Additionally, the proposed system uses a multi-stage architecture where unsupervised and supervised modules are combined.

In the present study the collection of documents used has been relatively limited, being approximately 1,000 texts from the Greek language. This number of texts has been chosen mainly to allow for a more extensive evaluation of the effectiveness of the different stages. Furthermore, the use of a collection in the Greek language is intended to evaluate the proposed method - and more generally the SOM model (which so far has been applied mainly to texts of the English language) - in a highly inflectional language.

To our knowledge, related studies have not been carried out by other researchers on datasets involving documents in the Greek language, nor does an established benchmark involving Greek texts exist. Thus, one of the aims in the future is to compare the proposed method to related algorithms such as the WEBSOM. Similarly, the method proposed in the present article is intended to be applied in the future to texts from other languages, such as English. This will allow this method to be compared directly in terms of results to other methods such as the WEBSOM.

In the present study, the aim is to organize a document collection with the minimum of supervision, using as input lemma-based features. The proposed method is expected to be combined in the future with related studies focusing on the use of grammatical and syntactic-based features to reflect the personal styles of authors [9][10]. It is expected that by combining these two distinct though related directions of research, a more effective document-organization system can be generated, which addresses more effectively the needs of the human user..

Acknowledgements

The authors would like to acknowledge the assistance of Ms. Marina Vassiliou, especially in linguistic topics associated with this study.

References

[1] T. Kohonen, S. Kaski, K. Lagus, Salojärvi, J. Honkela, V. Patero and A. Saarela "Self-Organisation of a Massive Document Collection." IEEE Transactions on Neural Networks, Vol. 11, No. 3, pp. 574-585, 2000.

[2] K. Lagus, S. Kaski and T. Kohonen "Mining Massive Document Collections by the WEBSOM Method." Information Sciences, Vol. 163, No. 1-3, pp.135-156, 2004.

[3] R.T. Freeman and H. Yin "Web content management by self-organization", IEEE Transactions on Neural Networks, Vol. 16, No 5, pp. 1256-1268, 2005.

[4] S. Kaski "Dimensionality Reduction by Random mapping: Fast Similarity Computation for Clustering." In Proceedings of IJCNN'98, International Joint Conference on Neural Networks, Vol. 1, pp. 413-418, 1998.

[5] A. Georgakis, C. Kotropoulos, A. Xafopoulos, and I. Pitas, "Marginal median SOM for document organization and retrieval", Neural Networks, Vol. 17, No. 3, pp. 365-377, 2004.

[6] D. Pullwitt, "Integrating Contextual Information to Enhance SOM-based Text Document Clustering." Neural Networks, Vol. 15, pp. 1099-1106, 2002.

[7] K. Linden "Evaluation of Linguistic Features for Word Sense Disambiguation with Self-Organised Document Maps". Computers and the Humanities, Vol. 38, pp. 417-435, 2004.

[8] G. Tambouratzis, "Assessing the effectiveness of Feature Groups in Author Recognition Tasks with the SOM Model." IEEE Transactions on Systems, Man & Cybernetics, Vol. 36, No. 2, pp. 249-259, 2006.

[9] G. Tambouratzis, S. Markantonatou, N. Hairetakis, M. Vassiliou, D. Tambouratzis and G. Carayannis "Discriminating the Registers and Styles in the Modern Greek Language – Part 2: Extending the Feature Vector to Optimise Author Discrimination", Literary and Linguistic Computing, Vol. 19, No. 2, pp. 221-242, 2004.

[10] G. Tambouratzis and M. Vassiliou "Employing Thematic Variables for Enhancing Classification Accuracy Within Author Discrimination Experiments." Literary and Linguistic Computing, Vol.22, No.2, pp.207-224, 2007.

[11] H. Papageorgiou, P. Prokopidis, V. Giouli and S. Piperidis, "A Unified PoS Tagging Architecture and its Application to Greek", Second International Conference on Language Resources and Evaluation Proceedings, Athens, Greece, Vol. 3, pp. 1455 – 1462, 2000.

[12] T. Kohonen, "Self-Organized formation of topologically correct feature maps", Biological Cybernetics, Vol. 43, pp. 59-69, 1982.

[13] J. Vesanto, J. Himberg, E. Alhoniemi and J. Parhankangas "SOM Toolbox for MATLAB 5". SOM Toolbox team, Helsinki University of Technology, Helsinki, Finland. [Online] Report A57, 2000.

[14] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organising Map". IEEE Transactions on Neural Networks, Vol. 11, No. 3, pp. 586-600, 2000.

[15] R.O. Duda, P.E. Hart and D.G. Stork, "Pattern Classification" (2nd edition), Wiley, New York, 2001.

[16] M. Riedmiller and H. Braun "A direct adaptive method for faster backpropagation learning: the RPROP algorithm", Proceedings of the IEEE International Conference on Neural Networks, San Francisco, pp. 586-591, 1993.

[17] C. Igel and M. Huesken "Empirical evaluation of the improved Rprop learning algorithm", Neurocomputing, Vol. 50, pp. 105-123, 2003.

[18] S. Haykin "Neural Networks: A comprehensive foundation", (2nd edition), Prentice Hall, 1999.

[19] D. Nguyen and B. Widrow "Improving the learning speed of 2-layer neural networks by choosing initial values of adaptive weights", Proceedings of the International Joint Conference on Neural Networks, Vol. 3, pp. 21-26, 1990.