International Journal of Engineering and Technology Volume 1 No. 2, November, 2011

Document Classification, a Novel Neural-based Classifier

Seyyed Mohammad Reza Farshchi

Islamic Azad University, Mashhad Branch, Dep. of Artificial Intelligence, Iran, Mashhad.
Shiveex@gmail.com, +989153081125.

ABSTRACT

The assignment of natural language texts to one or more predefined categories based on their content – is an important component in many information organization and management tasks. This research proposes a novel approach for documents classification with using novel method that combined competitive self organizing neural text categorizer with new vectors that we called, string vectors. Even if the research on document categorization has been progressed very much, documents should be still encoded into numerical vectors. Such encoding so causes the two main problems: huge dimensionality and sparse distribution. Although many various feature selection methods are developed to address the first problem, but the reduced dimension remains still large. If the dimension is reduced excessively by a feature selection method, robustness of document categorization is degraded. The idea of this research as the solution to the problems is to encode the documents into string vectors and apply it to the novel competitive self organizing neural text categorizer as a string vector. We compare the effectiveness of five different automatic learning algorithms for text categorization in terms of learning speed, real-time classification speed, and classification accuracy. The quantitative and qualitative experiment results demonstrate that this method can significantly improve the performance of documents classification.

Key words: Weil Text Classification (TC), Documents Classification, Information Management, Data Mining

1. INTRODUCTION

As the volume of information continues to increase, there is growing interest in helping people better find, filter, and manage these resources. Text categorization (TC a.k.a. Text classification, or topic spotting) - the assignment of natural language documents to one or more predefined categories based on their semantic content - is an important component in many information organization and management tasks [1].

Automatic text categorization task can play an important role in a wide variety of more flexible, dynamic and personalized tasks as well: real-time sorting of email or files, document management systems, search engines, digital libraries.

In the last 10 years content-based document management tasks (collectively known as information retrieval—IR) have gained a prominent status in the information systems field, due to the increased availability of documents in digital form and the ensuing need to access them in flexible ways [2].

TC the activity of labeling natural language texts with thematic categories from a predefined set, is one such task. TC dates back to the early 60's, but only in the early 90's did it become a major sub field of the information systems discipline, thanks to increased applicative interest and to the availability of more powerful hardware. TC is now being applied in many contexts, ranging from document indexing based on a controlled vocabulary [3], to document filtering, automated meta data generation, word sense disambiguation, population of hierarchical catalogues of Web resources, and in general any application requiring document organization or selective and adaptive document dispatching.

In many contexts trained professionals are employed to categorize new items. This process is very time-consuming and costly, thus limiting its applicability. Consequently there is an increasing interest in developing technologies for automatic text categorization [4].

A number of statistical classification and machine learning techniques has been applied to text categorization, including regression models, nearest neighbor classifiers, decision trees, Bayesian classifiers, Support Vector Machines (SVM), rule learning algorithms, relevance feedback, voted classification, and neural networks.

The research on text categorization has been made very much progress in context of machine learning and data mining. It requires encoding documents into numerical vectors for using one of traditional algorithms for text categorization [5].

A corpus which is a collection of documents is mapped into a list of words as the feature candidates. Among the candidates, only some are selected as the features. For each document, a numerical value is assigned to each of the selected features, depending on the importance and presence of each feature. However, encoding documents so causes the two main problems: huge dimensionality and sparse distribution [6].

In order to solve the two main problems, this research uses the novel method that documents should be encoded into string vectors. A string vector refers to a finite set of strings which are words in context of a natural language. In numerical vectors representing documents,

words are given as features, while in string vectors, words are given as feature values. Features of string vectors are defined very variously as properties of words with respect to their posting, lexical category, and statistical properties, but in this research, the highest frequent word, the second highest frequent one, and so on are defined as features of string vectors for easy implementation.

By encoding documents into string vectors, we can avoid completely the two main problems: huge dimensionality and sparse distribution.

We proposed the competitive neural text categorizer, as the approach to text categorization and proposed the application of it to documents categorization. Before creating the proposed neural network, traditional neural networks, such as MLP (Multi Layers Perceptron) with BP (Back Propagation) receives numerical vectors as its input data. Differently from the traditional neural networks, the proposed neural network receives string vectors. It has the two layers as its architecture: the input layer and the competitive layer. It is expected for the proposed model to improve the performance of text categorization by solving the two main problems.

The rest of this paper is organized as follows. The principle of TC and previous works is given in next sections. Strategies of encoding documents were given in sections 2. Section 3 describes the novel competitive self organizing neural text categorizer model. In section 4 we will mention the simulation result and significance of this research. Conclusions are presented in Section 5.

2. RELATED WORK

This section is concerned with previous works relevant to this research and we will survey previous relevant works, and point out their limitations. There exist other kinds of approaches to text categorization than machine learning based ones: heuristic and rule based approaches. Heuristic approaches were already applied to early commercial text categorization systems [7]. However, we count out the kind of approaches in our exploration, since they are rule of thumbs. Since rule based approaches have poor recall and require a time consuming job of building rules manually as mentioned in the previous section, they are not covered in this article, either. Therefore, this article counts only machine learning based approaches to text categorization considered as state of the art ones. Even if many machine learning approaches to text categorization already proposed, we will mention the four representative and popular approaches: KNN (K Nearest Neighbor), NB (Naive Bayes), SVM, and BP Neural Networks (NNBP or briefly BP) [8].

It requires encoding documents into numerical vectors for using one of them for text categorization; the two main problems are caused. String kernel was proposed in using the SVM for text categorization as the solution to the two main problems, but it failed to improve the performance [9]. In this section, we will explore the previous works on traditional approaches to text categorization and previous solution to the two main problems.

The KNN may be considered as a typical and popular approach to text categorization [10]. The KNN was initially created by Cover and Hart in 1967 as a genetic classification algorithm [11]. It was initially applied to text categorization by Massand et al at 1993 in [12]. The KNN algorithm is quite simple: given a test documents, and uses the categories of the K neighbors to weight the category candidates. The similarity score of each neighbor documents to test documents is used as the weight of the K nearest neighbor documents. If several of nearest neighbor share a category, then the per-neighbor weights of that category are added together, and the resulting weighted sum is used as the like hood score of that category with respect to the test document. By scoring the scores of candidate categories, a ranked list is obtained for the rest document. KNN was recommended by Yang at 1999 in [13] and by Sebastiani at 2002 in [14] as a practical approach to text categorization. Therefore, the KNN has been aimed as the base approach in other literature as the base approach. The Naive Bayes may be considered as another approach to text categorization. It was initially created by Kononenko in 1989, based on Bayes Rule [15]. Its application to text categorization was mentioned in the textbook by Mitchell in [16]. Assuming that the Naive Bayes is the popular approach, in 1999, Mladenic and Grobelink proposed and evaluated feature selection methods [17]. The Naive Bayes has been compared with other subsequent approaches in text categorization at [18].

Recently, the SVM was recommended as the practical approach to text categorization [19]. It was initially introduced in Hearst magazine in [20]. In the same year, it was applied to text categorization by Joachims [21]. Its idea is derived from a linear classifier perceptron, which is an early neural network. Since the neural network classifies objects by defining a hyper-plane as a boundary of classes, it is applicable to only linearly separable distribution of training examples. The main idea of SVM is that if a distribution of training examples is not linearly separable, these examples are mapped into another space where their distribution is linearly separable, as illustrated in the left side of figure 1. SVM optimizes the weights of the inner products of training examples and its input vector, called Lagrange multipliers [22], instead of those of its input vector, itself, as its learning process. In fact, the method is defined over a vector space where the problem is to find a decision surface that "best" separate the data points in two classes. In order to define the "best" separation, we need to introduce the "margin" between two classes. Figure 2 and 3 illustrate the idea. For simplicity, we only show a case in a two dimensional space with linearity separable data points. It was adopted as the approach to spam mail filtering as a practical instance of text categorization by Druker et al in [23]. Furthermore, the SVM is popularly used not only for text categorization

tasks but also for any other pattern classification tasks [24].

In 1995, BP was initially applied to text categorization by Wiener in his master thesis [25]. He used Reuter 21578 [26] as the test bed for evaluating the approach to text categorization and shown that back propagation is better than KNN in the context of classification performance. In 2002, Gabriel applied continually BP to text categorization [27]. They used a hierarchical combination of BPs, called HME (Hierarchical Mixture of Experts), to text categorization, instead of a single BP. They compared HME of BPs with a flat combination of BPs, and observed that HME is the better combination of BPs. Since BP learns training examples very slowly, it is not practical, in spite of its broad applicability and high accuracy, for implementing a text categorization system where training time is critical.

Research on machine learning based approaches to text categorization has been progressed very much, and they have been surveyed and evaluated systematically. In 1999, neural networks may be considered as an approach to text categorization, and among them, the MLP with BP is the most popular model [28].

The neural network model was initially created in 1986 by Mcelland and Rumelhart, and it was intended to performing tasks of pattern classification and nonlinear regressions as a supervised learning algorithm [29]. It was initially applied to text categorization in 1995 by Wiener [25]. Its performance was validated by comparing it with KNN in his master thesis on the test bed, Reuter21578. Even if the neural network classifies documents more accurately, it takes very much time for learning training documents.

The string kernel was proposed as the solution to the two main problems which is inherent in encoding documents into numerical vectors. It was initially proposed by Lodhi et al in 2002 as the kernel function of SVM [31]. String kernel receives two raw texts as its inputs and computes their syntactical similarity between them. Since documents don't need to be encoded into numerical vectors, the two main problems are naturally avoided. However, it costed very time for computing the similarity and failed to improve the performance of text categorization.

This research has three advantages as mentioned in this section. The first advantage of this research is to avoid the two main problems by encoding the documents into alternative structured data to numerical vectors. The second advantage is that string vectors are more transparent than numerical vectors with respect to the content of its full text; it is easier to guess the content of document by seeing its string vector than by its numerical vector specially when we want to classify some documents such Persian documents (Persian data are more complex). The third advantage as one derived from the second advantage is that it is potentially easier to trace why each document is classified. Therefore, this research proposes the novel method that creates a competitive self organizing neural network which received string vectors of documents data as its input data because of the three advantages.

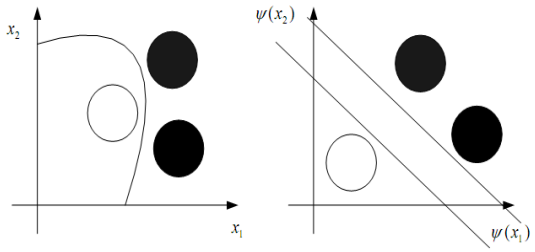

Figure 1: Mapping vector space in SVM

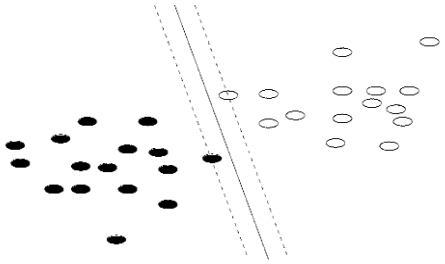

Figure 2: A decision line (solid) with a smaller margin

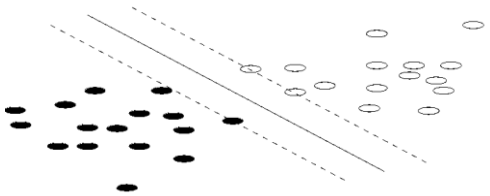

Figure 3: A decision line (solid) with the maximal margin

3. STRATEGIES OF ENCODING DOCUMENTS

Since the documents are unstructured data by themselves they cannot be processed directly by computers. They need to be encoded into structured data for processing them for text categorization. This section will describe the two strategies of encoding: the traditional strategy and the proposed strategy. The first subsection describes the formal description of TC, then the former, points out the two strategies of encoding documents.

3.1 Formal Description of TC Problem

Categorization is the task of assigning a Boolean value to each pair $< d_j, c_i > \in D \times C$ where D is a domain of documents and $C = \{c_1, c_2, ..., c_{|C|}\}$ is a set of predefined categories. A value of T assigned to $< d_j, c_i >$ indicates a decision to file d_j under c_i while a

value of F indicates a decision not to file d_j under c_i. More formally the task is to approximate the unknown target function $\Phi := D \times C \rightarrow \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\Phi := D \times C \rightarrow \{T, F\}$, called the classifier.

4. COMPETITIVE LEARNING

In competitive learning, the output neurons of a neural network compete among themselves to become active (to be "fired"). Whereas in MLP several output neurons may be active simultaneously, in competitive learning only a single output neuron is active at any time.

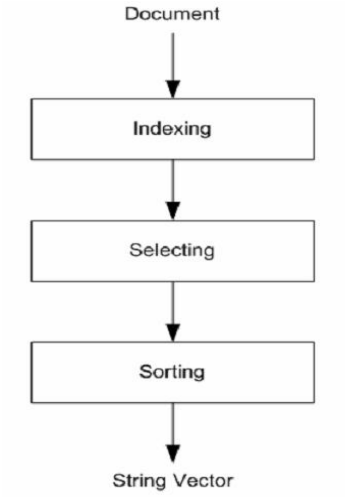

Figure 4: The process of mapping documents into a string vector

In other words, competitive learning is a learning procedure that divides a set of input patterns in clusters that are inherent to the input data. A competitive learning network is provided only with input vectors *x* and thus implements an unsupervised learning procedure.
A simple competitive learning network was depicted in Figure 5. A basic competitive network has an input layer and a competitive layer. The nodes in the competitive layer "compete" with each other, and the node that has the largest output becomes the "winning" neuron. The winning neuron is set to 1 and all other neurons are set to 0.

The training of the basic competitive network uses the Kohonen learning rule. For each input pattern, the weight vector of the winning node is moved closer to the input vector using the following formula:

$$w_i(q) = w_i(q-1) + \alpha(p(q) - w_i(q-1)) \qquad (2)$$

Where w_i is the weight of the winning neuron, *p* is the corresponding input vector (string value) and *D* is the Kohonen learning rate. However, a problem of this model is that if the initial weight of a neuron is far from any vector, it will never be trained, so a bias vector is added to the result of the competition. The winning node would cause the bias vector to decrease. Under this mechanism, it is more difficult for a neuron to continue to win. The degree of bias is represented by a factor called conscience rate. As we show in figure 6 each of the four outputs *O* is connected to all inputs *i* with weight w_{i0}. When an input string vector *x* is presented only a single output unit of the network (the winner) will be activated. In a correctly trained network, all *x* in one cluster will have the same winner. For the determination of the winner and the corresponding learning rule, two methods exist: dot products and Euclidean distance. For simplicity of calculation we used the Euclidean distance in proposed network.

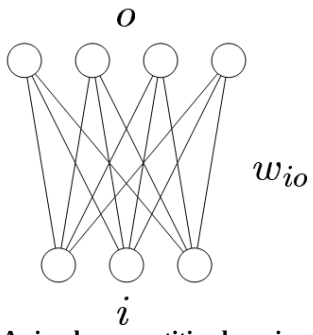

Figure 5: A simple competitive learning network

The proposed neural network follows self organizing map (SOM) in that synaptic weights are connected directly between the input layer and the competitive layer, and the weights are updated only when each training example is misclassified.

However, note that the proposed neural network is different from SOM in context of its detail process of learning and classification, since it uses string vectors as its input vectors, instead of numerical vectors. The competitive layer given as an additional layer to the input layer is different from the hidden layer of back propagation with respect to its role. The learning layer determines synaptic weights between the input and the competitive layer by referring to the tables owned by learning nodes. The learning of proposed neural network refers to the process of competition between weights stored in the tables.

Each training example is classified by summing the initial weights and selecting the category corresponding to the maximal sum. If the training example is classified correctly, the weights are not updated. Otherwise, the weights are incremented toward the target category and those are decremented toward the classified category. The winner weights (target category) are generated as the output of this process.

In the competitive neural network, each example is classified by summing the winner optimized weights, whether it is a training or unseen example. In addition weights connected to itself from the input nodes as its categorical score. The weights are decided by referring the

table which is owned by its corresponding learning node. The category corresponding to the output node which generate its maximum categorical score (winner category) is decided as the category of the given example. Therefore, the output of this process is one of the predefined categories, assuming that the competitive neural network is applied to text categorization without the decomposition. Figure 6 shows the diagram of proposed neural network. Complete algorithm of competitive neural text classifier and competitive learning algorithm was mentioned in classifier training algorithm and learning algorithm, respectively.

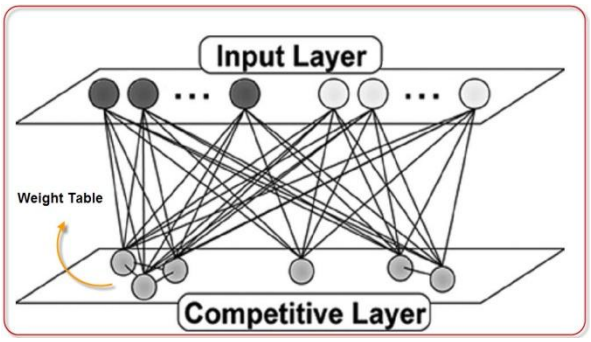

Figure 6: Proposed self organizing neural text classifier

Algorithm 1: Classifier Training

Input: A Series of Documents, Number of Categories
Output: The Winner Categories

1: Encode these sample documents into string vectors
2: Design the architecture of competitive text categorizer
3: Initialize weights in each learning node in competitive layer with its document
4: Repeat step 1-3 with the number of given documents
5: For each encoded sample document
6: Compute the values of winner nodes of the encoded
6-1: Classify each training vector into the corresponding category
6-2: Output: winner node in each learning node
6-2-1: If the winner node classify the documents correctly go to step 7
6-2-2: Update table weights
7: Output calculated weights
8: End

Algorithm 2: Winner Selection

Input: The Architecture of competitive self organizing neural text categorizer
Output: Selected Winner

1: Given one of string vector that in previous stage was created.
2: Compute the output value of nodes in the encoded document using the equation (2).
3: Classify the unseen string vector into the category corresponding to the output node (winner node)
4: End

5. EXPERIMENTAL CONSIDERATION

This section is concerned with the empirical validation of the performance of proposed method in several experiments.

An important issue of text categorization is how to measure the performance of the classifiers. Many measures have been used, each of which has been designed to evaluate some aspect of the categorization performance of a system [28]. In this section we discuss and analysis the important measures that have been reported in the literature.

We use the collection of Persian news categories, called irna.ir. In addition, For evaluating our method on English documents the standard test bed, Reuter 21578, was used. The Reuter 21578 is popularly used as the standard test bed for evaluating approaches to text categorization.

This set of experiments involves the five approaches: KNN, NB, SVM, NNBP, and our proposed method. In experiment result, the test bed and configurations of the approaches involved in the set of experiments are described, and the results of the set of experiments are presented and discussed.

The partition of the test bed, Reuter 21578 and irna.ir into the training and test set is illustrated in table 3 and 4, respectively. The test bed contains the most frequent categories of different type of news for entering the first evaluation, and its source is the web site, www.irna.ir. The collection was built by copying and pasting the news documents individually as the plain text files. In the test bed, the five categories and the 5,436 Persian and English news documents are available. The collection of news articles is partitioned into the training and test set by the ratio 7:3, as shown in table 3 and 4.

Table 3: Collection of different news articles on Reuter 21578

Category Name	Training Set	Test Set	Total
Trade	869	380	575
Earn	500	214	515
Grain	220	94	245
Wheat	430	185	615
Ship	250	110	360
Corn	280	120	400
Total	1890	820	2710

The configurations of the involved approaches are illustrated in table 5. The parameters of the SVM and the

KNN, the capacity and the number of nearest neighbors, are set as five and six, respectively, but the NB has no parameter. The parameters of the NNBP such as the number of hidden nodes And the learning rate are arbitrary set as shown in table 5.

Persian news documents are encoded into 420 dimensional numerical vectors and 123 dimensional string vectors. English documents are encoded into 398 numerical vectors and 116 dimensional string vectors. We compared performance of the proposed method with four traditional approaches in following experiments.

5.1 Micro and Macro Averaging

For evaluating performance average across categories, there are two conventional methods, namely macro-averaging and micro-averaging. Macro-averaged performance scores are determined by first computing the performance measures per category and then averaging these to compute the global means. Micro-average performance scores are determined by first computing the totals of *a*, *b*, *c*, and *d* for all categories and then use these totals to compute the performance measures. There is an important distinction between the two types of averaging. Micro-averaging gives equal weight to every document, while macro-averaging gives equal weight to each category. For evaluating the performance of the classifiers, we define four parameters:

- *a* - The number of documents correctly assigned to this category.
- *b* -The number of documents incorrectly assigned to this category.
- *c* - The number of documents incorrectly rejected from this category.
- *d* - The number of documents correctly rejected from this category.

The results of this experiment on Reuter 21578 test bed are presented in figure 8. Among the five methods, the left picture indicates the micro-averaged measure of each method. The right picture indicates the macro-averaged measure of each method, respectively. Our proposed approach shows its best performance to the NNBP, but the performance of our proposed approach is comparable to that of NNBP.

Table 4: Collection of different news articles on irna.ir

Category Name	Training Set	Test Set	Total
Politics	350	175	525
Law	360	145	505
Computer	150	75	225
Education	110	47	157
Economics	472	203	675
Sports	466	200	666
Total	1908	845	2753

Let's discuss the results from the set of experiments which were illustrated in figure 8. Even if the macro-averaged proposed neural network is not better than NNBP in the task, both are comparable to each other with respect to the performance of text categorization. Note that it requires very much time for training NNBP as the payment for the best performance. In addition, the NNBP is not practical in dynamic environments where NNBP must be trained again, very frequently. Hence, the proposed method is more recommendable than NNBP with respect to both the learning time and the performance.

5.2 F-Measure

Another evaluation criterion that combines recall and precision is the F-measure. In fact, the F1 measure is used for evaluating the performance of TC. The F1measure can be calculated as following equation:

$$E(P) = \sum \frac{(1+\beta) \times \mathrm{Re}call\,(i,k) \times precision(i,k)}{\beta \times \mathrm{Re}call\,(i,k) + precision(i,k)} \quad (3)$$

Precision and recall are widely used for evaluation measures in TC. For calculating the F1 measure, in each category and each documents we should determines whether the document belongs to the category or not. So we need to define the recall and precision rate with the parameters that defined in previous section as:

$$recall = \frac{a}{a+c}$$
$$precision = \frac{a}{a+b} \quad (4)$$

Table 5: Parameter settings of algorithms

Algorithms	Parameter Settings
SVM	Capacity = 5.0
KNN	# Nearest Value= 6
Naïve Bayes	N/A
NN With Back Propagation (BP)	# Hidden Layer=15 Learning Rate=0.2 #Training Epoch=500

Algorithms	Parameter Settings
Proposed Method	Learning Rate=0.2 #Training Epoch=150

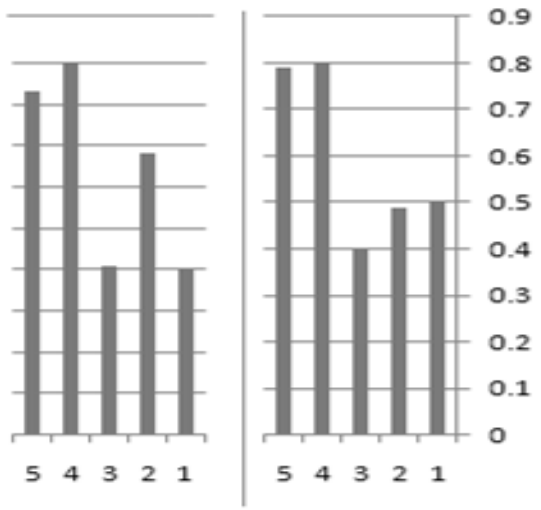

Figure 7: The left side of figure indicates the micro-averaged measure and the right one shows the macro-averaged for (left to right): proposed method, NNBP, SVM, NB, KNN

Figure 8 shows the result of evaluating the F1 measure for five approaches on the irna.ir test bed. Science each category contain identical number of test documents, micro-averaged and macro-averaged F1 are same as each other. Therefore, their performances are presented in an integrated group, instead of two separated groups, in figure 8. This result shows that back propagation is the best approach in comparison to another three traditional algorithms, while NB is the worst approach with the decomposition of the task on this test bed. Unlike the previous experiment set, NTC is comparable and competitive with back propagation. So we discuss this analysis in next subsections with combined to another experiments

5.3 Accuracy

Figure 10 show the accuracy of all methods on Reuter 21578 news document test bed. This picture show that the proposed neural network has more reliable than other traditional method.

The accuracy rate of the proposed neural network on test bed is more than 86% but the best traditional approach have 80% accuracy rate.

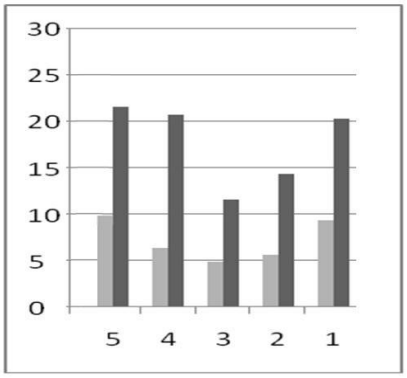

Figure 8: The F1 measure evaluation for (left to right): proposed method, KNN, NB, SVM, NNBP

5.4 Recall and Precision Rate

We also tried another performance measure for our proposed method to show the quality of document classification. We validate the performance of novel approach by comparing it with other machine learning algorithms on the irna.ir test bed in this experiment. Table 7 shows these rate for best traditional method and novel method.

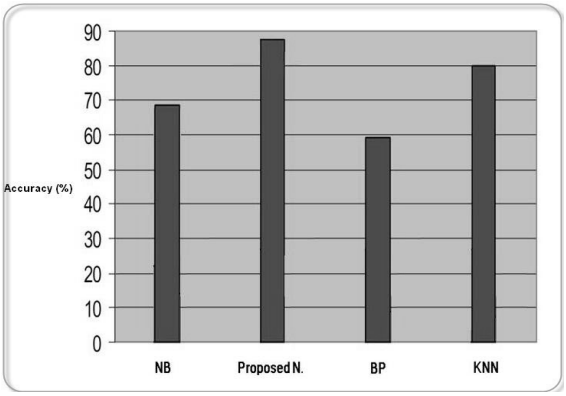

Figure 10: Accuracy rate of news classifier on Reuter 21578

From the above result (and section 4.2) we can see that the documents classifier based on competitive self organizing neural network with string vector can classify document of different categories correctly, represented by a high accuracy rate. If we use the positive and negative accuracy rate for evaluating the performance of proposed text classifier we have:

Table 6: Positive and negative accuracy for news text classifier

Positive Accuracy	Negative Accuracy	Average Accuracy
0.4894	0.9368	0.7131

Where a low positive accuracy rate shows that many documents from different categories are not clustered together. This is partly because it is difficult to

have information about the correct category in unsupervised learning. In addition this result shows that the classifiers based on BP network (NNBP) couldn't classify documents (in most categories) correctly.

Table 7: Precision and Recall rate of best traditional and novel text classifier

	Precision	Recall
SVM	0.6398	0.4
NNBP	0.4367	0.4
KNN	0.5612	0.8
NB	0.7866	0.65
Our Method	0.9107	0.9

In SVM experiment, precision and recall are low in some categories. The novel method takes less than one-tenth of the time BP takes when training. At the same time, it performs well in the categories in which the results are satisfactory. In the different news category, it even outperforms the NB method. Figure 10 shows the complete recall and precision rate on some category on irna.ir test bed. This picture shows the robustness and quality of text categorization by the competitive self organizing neural text categorizer. The novel method can over perform the traditional method with classify precision rate of 0.8.

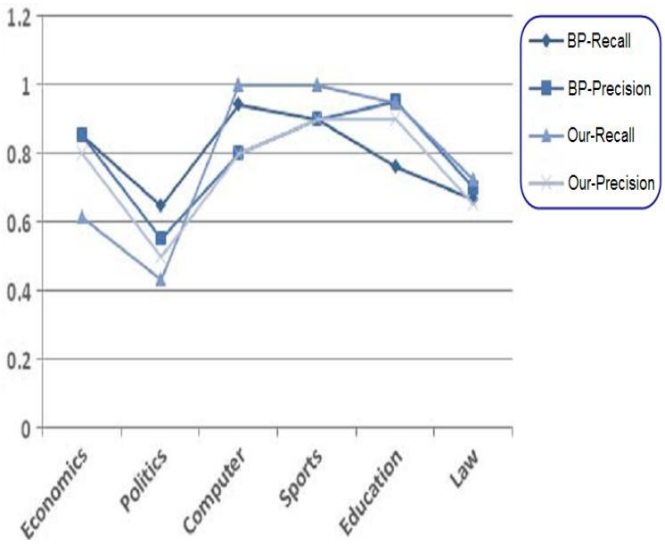

Figure 10: Precision and Recall rate between best traditional and novel algorithm on each categories

6. CONCLUSION

This research proposes a novel method that used competitive self organizing neural network with string vector for text categorization which uses alternative representations of documents to numerical vectors. In this method we used a full inverted index as the basis for the operation on string vectors, instead of a restricted sized similarity matrix. It was cheaper to build an inverted index from a corpus than a similarity matrix, as mentioned in section 2. In the previous attempt, a restricted sized similarity matrix was used as the basis for the operation on string vectors. Therefore, information loss from the similarity matrix degraded the performance of the modified version. This research addresses the information loss by using a full inverted index, instead of a restricted sized similarity matrix.

The four contributions are considered as the significance of this research. For first, this research proposes the practical approach for documents categorization, according to the results of the set of experiments. For second, it solved the two main problems, the huge dimensionality and the sparse distribution which are inherent in encoding documents into numerical vectors. For third, it created a new neural network, called competitive self organizing neural text categorizer, which receives string vectors differently from the previous neural networks. For last, it provides the potential easiness for tracing why each news document is classified so. Other machine learning algorithms such as Naïve Bayes and back propagation are considered to be modified into their adaptable versions to string vectors. The operation may be insufficient for modifying other machine learning algorithms. For example, it requires the definition of a string vector which is representative of string vectors corresponding to a mean vector in numerical vectors for modifying k-means algorithm into the adaptable version. Various operations on string vectors should be defined in a future research for modifying other machine learning algorithms.

Let's consider another remaining task as the further research. The first task is to apply the proposed competitive self organizing neural network to categorization of documents within a specific domain such as medicine, law, and engineering. The second task is to modify it into the static version.

REFERENCES

[1] K. Androutsopoulos, K. V. Koutsias, Chandrinos, C. D. Spyropoulos, "An Experimental Comparison of Naïve Bayes and Keyword-based Anti-spam Filtering with Personal Email Message", Proceedings of 23rd ACM SIGIR, pp.160-167, 2000.

[2] N. Cristianini, J. Shawe-Taylor, "Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000.

[3] N.L. Bhamidipati, S.K. Pal, "Stemming via Distribution-Based Word Segregation for

Classification and Retrieval", IEEE Transactions on Systems Man. and Cybernetics, Vol.37, No.2, pp.350-360, 2007.

[4] R.O. Duda, P.E. Hart, D.G. Stork, "Pattern Classification", John Wiley & Sons, Inc, 2001.

[5] V.I. Frants, J. Shapiro, V.G. Voiskunskii, "Automated Information Retrieval: Theory and Methods", Academic Press, 1997.

[6] M.T. Hagan, H.B. Demuth, M. Beale, "Neural Network Design", PWS Publishing Company, 1995.

[7] S. Haykin, "Neural Networks: Comprehensive Foundation", Macmillan College Publishing Company, 1994.

[8] A. Frolov, D. Husek, "Recurrent Neural Network based Boolean Factor Analysis and Its Application to Word Clustering", IEEE Transactions on Neural Networks, Vol.20, No.7, pp.1073-1086, 2009.

[9] P. Jackson, I. Mouliner, "Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization", John Benjamins Publishing Company, 2002.

[10] T. Martin, H.B. Hagan, H. Demuth, M. Beale, "Neural Network Design", PWS Publishing Company, 1995.

[11] L. Man, S. Jian, "Empirical Investigations into Full-Text Protein Interaction Article Categorization Task (ACT) in the Procreative II Challenge", IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol.7, No.3, pp.421-427, 2010.

[12] B. Massand, G. Linoff, D. Waltz, "Classifying News Stories using Memory based Reasoning", Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval, pp.59-65, 1992.

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

[14] F. Sebastiani, "Machine Learning in Automated Text Categorization", ACM Computing Survey, Vol.34, No.1, pp.1-47, 2002.

[15] J. Rennie, "Improving Multi-class Text Classification with Support Vector Machine", Master's thesis, Massachusetts Institute of Technology, 2001.

[16] T. M. Mitchell, "Machine Learning", McGrawHill, 1997.

[17] D. Mladenic, M. Grobelink, "Feature Selection for Unbalanced Class Distribution and Naïve Bayes", Proceedings of International Conference on Machine Learning, pp.256-267, 1999.

[18] M.E. Ruiz, P. Srinivasan, "Hierarchical Text Categorization Using Neural Networks", Information Retrieval, Vol.5, No.1, pp.87-118, 2002.

[19] J.C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, 1998.

[20] M. Hearst, "Support Vector Machines", IEEE Intelligent Systems, Vol.13, No.4, pp.18-28, 1998.

[21] T. Joachims, "Text Categorization with Support Vector Machines: Learning with many Relevant Features", The Proceedings of 10th European Conference on Machine Learning, pp.143-151, 1998.

[22] D.A. Bell, J.W. Guan, "On Combining Classifier Mass Functions for Text Categorization", IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.10, pp.1307-1319, 2005.

[23] H. Drucker, D. Wu, V.N. Vapnik, "Support Vector Machines for Spam Categorization", IEEE Transaction on Neural Networks, Vol.10, No.5, pp.1048-1054, 1999.

[24] P.G Espejo, S. Ventura, "A Survey on the Application of Genetic Programming to Classification", Systems, IEEE Transactions on Man. and Cybernetics, Vol.40, No.2, pp.121-144, 2010.

[25] E.D. Wiener, "A Neural Network Approach to Topic Spotting in Text", The Thesis of Master of University of Colorado, 1995.

[26] http://www.research.att.com/~lewis/reuters21578.html

[27] F. Gabriel Pui Cheong, J. Yu, "Text Classification without Negative Examples Revisit", IEEE Transactions on Knowledge and Data Engineering, Vol.18, No.1, pp.6-20, 2006.

[28] D. Isa, L. Lee, "Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine", IEEE Transactions on Knowledge and Data Engineering, Vol.20, No.9 pp.1264-1272, 2008.

[29] V. Lertnattee, T. Theeramunkong, "Multidimensional Text Classification for Drug Information", IEEE Transactions on Information Technology in Bio medicine, Vol.8, No.3, pp.306-312, 2008.

[30] L. Man, Chew, T. Lim, "Supervised and Traditional Term Weighting Methods for Automatic Text Categorization", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.31, No.4, pp.721-735, 2009.

[31] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins, "Text Classification with String Kernels", Journal of Machine Learning Research, Vol.2, No.2, pp.419-444, 2002.

ABOUT AUTHOR

S.M.R Farshchi was born in Mashhad, Iran, 1988. He received the Ms degree in Artificial Intelligence from the Islamic Azad University, Mashhad Branch, Iran, 2010. He was a Research Scientist with the Institute of Iran Cognitive Science Laboratory from 2006 to 2010 and with the National Institute of Advanced Industrial Science and Technology, Mashhad, from 2006 to 2010. From 2008 to 2010, he was a Visiting Scientist with the Machine Learning in the Institute of Sadjad University . In Oct. 2009, he became a Faculty Member with the Imaging Science and Engineering Laboratory, Sadjad Institute of Technology, Mashhad, Iran.