Teaching Neural Networks to Detect the Authors of Texts Using Lexical Descriptors

Mehmet Can[‡], Amir Jamak[§]
and Alen Savatic[**]

Abstract

This paper proposes a means of using an artificial neural network to distinguish the authors of paragraphs. Once the network has been trained, its hidden layer activations are recorded as a representation of the average number of words and average characters of words in a paragraphs of an author. This stored information can then be used to identify the texts written by authors. This computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks. Computational simplicity is achieved by distributing the learning task among a number of experts, which in turn divides the input space into a set of subspaces. The combination of these experts is said to constitute a committee machine. Basically, it fuses knowledge acquired by experts to arrive at an overall decision that is supposedly superior to that attainable by anyone of them acting alone. By this, we succeeded to distinguish the paragraphs authored by Ivo Andrić, from the ones authored by Mehmed Meša Selimović.

Keywords—Machine learning, author identification, artificial neural networks

[*]Emeritus Prof. International University of Sarajevo, Faculty of Engineering and Natural Sciences Sarajevo,Bosnia and Herzegovina.
[§] International University of Sarajevo, Faculty of Engineering and Natural Sciences, Bosnia and Herzegovina
[**] International University of Sarajevo, Faculty of Engineering and Natural Sciences, Sarajevo,Bosnia and Herzegovina

1. INTRODUCTION

Author identification denotes quantitative analysis of some written text that yields information about the style it is composed with and through that about the author of this text. The main author identification tasks are author characterization, similarity detection, and author identification [1].

Author characterization brings conclusions about the author, such as gender, education, social background etc. Similarity detection involves comparing texts of several authors in order to find, if they exist, some properties in common in the texts of the same author, or different authors. Author identification means attributing an unknown text to a writer based on some feature characteristic or measure. It can be used when several people claim to have written some text or when no one is able or willing to identify the real author of this text.

Stylometry is most often used for detection of plagiarism, finding authors of anonymously published texts, for disputed authorship of literature or in criminal investigations within forensic linguistic domain.

Two critical issues of the author identification analysis are: selection of descriptors that characterize texts and authors, and analytical techniques and algorithms applied to the task.

The typical textual analysis procedure starts with training during which there are used texts of known authors for whom there are computed characteristics of selected features, then follows the stage of verification when for unattributed texts there are obtained the same descriptors to be compared with previously calculated results. Then from the available set of possible authors there is chosen the one that matches most closely.

Features selected [2] in author identification methods must constitute the author's invariant properties of texts which is an invariant of its author, that is it is similar in all texts of this author and different in texts of different authors. It is generally agreed that writer invariants exist yet establishing what properties of a text should be used is an open question [3].

Usually analytical techniques applied to author identification tasks employ either statistic or machine learning approaches. Statistical computations are used in Markov models, principal component and linear discriminant analysis, clustering analysis, cumulative sum. Machine learning involves application of artificial neural networks, genetic algorithms, support vector machines [4], rough set theory, decision trees, and other similar methods.

In this paper an application to artificial neural networks is presented to authorship attribution is considered as a classification task [5]. Texts studied are literary works of two Bosnian writers, Ivo Andrić (1892-1975) and M. Meša Selimović (1910-1982). Feature selected

to describe texts are lexical and syntactical components that show promising results when used as writer invariants because they are used rather subconsciously and reflect the individual writing style which is difficult to be copied. Properly trained neural networks possess generalization properties that allow for the required high accuracy of classification.

2. OBJECTIVES OF AUTHOR IDENTIFICATION

The primary aim of author identification is to remove uncertainty about the author of some text, which can be used in literary tasks of textual analysis for works edited, translated, with disputed authorship or anonymous, but also with forensic aspect in view to detect plagiarism, forgery of the whole document or its constituent parts, verify ransom notes, etc.

Author identification analysts claim that each writer possesses some unique characteristic, called the authorial or writer invariant, that keeps constant for all texts written by this author and perceivably different for texts of other authors. To find writer invariants there are used style markers which are based on textual properties belonging to either of four categories: lexical, syntactic, structural, and content-specific [6].

Lexical descriptors provide statistics of total number of words or characters, average number of words per sentence, characters per sentence or characters per word, frequency of usage for individual letters or distribution of word length.

Syntactic features reflect the structure of sentences, which can be simple or complex, or conditional, built with punctuation marks. Structural attributes express the organization of text into paragraphs, headings, signatures, embedded drawings or pictures, and also special font types or its formatting that go with layout.

Content-specific properties recognize some keywords: words of special meaning or significant importance for the given context.

Unfortunately, the convenience of using contemporary word editors and processors works against preserving individual author styles due to its available options of "copy and paste". It makes imitation of somebody else's style much easier and that is why modern author identification techniques aim at exploiting the computational powers of computers to analyze patterns within subconsciously used common parts of speech, as opposed to historical approaches that emphasized some rare standing out elements of a text which could be noticed by virtually anybody and thus likely to be faked.

2.1 Historical View

Author identification evolved mainly from historical textual analysis methods dedicated to proving or disproving authenticity of documents or settling questions of authorial identity for anonymous or disputed texts.

As early as in 1439 Lorenzo Valla proved the forgery of the Donation of Constantine by comparing the Latin used in other documents dated to 4th Century that were unquestionably original [6].

Yet these early attempts could hardly rely on anything else but striking elements of texts such as distinct vocabulary or specific language structures [7].

The new era for author identification downed in 1887 when Mendenhall proposed to use not qualitative but quantitative measures such as word length, its average and distribution. This was followed by Yule and Morton, in 1938 and 1965, who selected sentence length as descriptive feature for authorship identification [8].

Numerical measurements of texts were not fully exploited at first but the development of computers with their high and permanently increasing computational powers made possible the application of statistical-oriented analysis to constantly growing corpus of texts in the cyberspace of Internet, enabling also to employ algorithms from machine learning domain to author identification tasks.

2.2. Methodologies employed

Contemporary author identification procedures are typically representatives of either computer-aided statistic based analysis, or artificial intelligence techniques.

In statistical analysis there are used computations of probabilities and distributions of occurrences for single letters or other characters such as punctuation marks, words, patterns of words or sentences [9].

One such method calculates the cumulative sum for two textual features. The first of these is the sentence length whose deviations from the average are plotted as the graph for the whole text sample of some known author. As the second descriptor typically there is chosen the usage of the 2 or 3 letter words, words starting with a vowel, or the combination of these two together. The two descriptors reflect the writing habits and are the key to detecting the author. If the two graphs match, the writer is identified [10].

Markov models consider a text as a sequence of characters (letter, punctuation marks, spaces, etc.) that corresponds to a Markov chain [11]. In probabilistic model of natural language letters appear with some probability, depending on which characters precede them. In the simplest model there is considered only the immediate predecessor which gives rise to the 1st order Markov chain. Thus for all pairs of letters in the alphabet there are obtained matrices of transition frequencies of one letter into another. These statistics are calculated for all texts by known authors and for some unattributed text as the true author there is selected the one with the highest probability.

Methods such as Linear Discriminant Analysis, Principal Component Analysis

or cluster analysis aim to reduce the dimensionality for input data and if procedures applied to texts of both known and unknown authors give the same result, the question of authorship identification is settled.

Genetic Algorithms provide an example of artificial intelligence technique applied in author identification analysis. The whole procedure starts with definition of a set of rules describing textual properties. Next these rules are checked against the text of known authorship and each rule if evaluated for fitness, basing on which score some rules (with the lowest score) are discarded leaving only these with fitness satisfying some criterion (selection process). The selected rules are slightly modified (mutation) and some new added, after which they are tested again. The process continues till there is obtained some number of rules that best describe features of the known text. At this point the evolved rules can be tested on a text of unknown author and if their fitness remains the same, the author is found.

Artificial Neural Networks are well suited to classification tasks by their ability to deal efficiently with large amount of data, especially in continuous domain since they do not require discretisation as for example classical rough sets. As the processing engine applied to research this paper presents, ANN with their architectures and training methods are described in the next section with more detail.

3. ARTIFICIAL NEURAL NETWORKS

Nervous systems existing in biological organism for years have been the subject of studies for mathematicians who tried to develop some models describing such systems and all their complexities. Artificial Neural Networks emerged as generalizations of these concepts with mathematical model of artificial neuron due to McCuloch and Pitts [12] described in 1943 definition of unsupervised learning rule by Hebb [13] in 1949, and the first ever implementation of Rosenblatt's perceptron [14] in 1958. The efficiency and applicability of artificial neural networks to computational tasks have been questioned many times, especially at the very beginning of their history the book "Perceptrons" by Minsky and Papert [15], published in 1969, caused dissipation of initial interest and enthusiasm in applications of neural networks. It was not until 1970s and 80s, when the backpropagation algorithm for supervised learning was documented that artificial neural networks regained their status and proved beyond doubt to be sufficiently good approach to many problems. Artificial Neural Network can be looked upon as a parallel computing system comprised of some number of rather simple processing units (neurons) and their interconnections. They follow inherent organizational principles such as the ability to learn and adapt, generalization, distributed knowledge representation, and fault tolerance. Neural network specification comprises definitions of the set of neurons (not only their number but also their organization), activation states for all neurons expressed by their activation

functions and offsets specifying when they fire, connections between neurons which by their weights determine the effect the output signal of a neuron has on other neurons it is connected with, and a method for gathering information by the network that is its learning (or training) rule.

3.1. Architecture

From architecture point of view neural networks can be divided into two categories: feed-forward and recurrent networks. In feed-forward networks the flow of data is strictly from input to output cells that can be grouped into layers but no feedback interconnections can exist. On the other hand, recurrent networks contain feedback loops and their dynamical properties are very important.

The most popularly used type of neural networks employed in pattern classification tasks is the feedforward network which is constructed from layers and possesses unidirectional weighted connections between neurons. The common examples of this category are Multilayer Perceptron or Radial Basis Function networks, and committee machines.

Multilayer perceptron type is more closely defined by establishing the number of neurons from which it is built, and this process can be divided into three parts, the two of which, finding the number of input and output units, are quite simple, whereas the third, specification of the number of hidden neurons can become crucial to accuracy of obtained classification results.

The number of input and output neurons can be actually seen as external specification of the network and these parameters are rather found in a task specification. For classification purposes as many distinct features are defined for objects which are analyzed that many input nodes are required. The only way to better adapt the network to the problem is in consideration of chosen data types for each of selected features. For example instead of using the absolute value of some feature for each sample it can be more advantageous to calculate its change as this relative value should be smaller than the whole range of possible values and thus variations could be more easily picked up by Artificial Neural Network. The number of network outputs typically reflects the number of classification classes.

The third factor in specification of the Multilayer Perceptron is the number of hidden neurons and layers and it is essential to classification ability and accuracy. With no hidden layer the network is able to properly solve only linearly separable problems with the output neuron dividing the input space by a hyperplane. Since not many problems to be solved are within this category, usually some hidden layer is necessary.

With a single hidden layer the network can classify objects in the input space that are sometimes and not quite

formally referred to as simplexes, single convex objects that can be created by partitioning out from the space by some number of hyperplanes, whereas with two hidden layers the network can classify any objects since they can always be represented as a sum or difference of some such simplexes classified by the second hidden layer.

Apart from the number of layers there is another issue of the number of neurons in these layers. When the number of neurons is unnecessarily high the network easily learns but poorly generalizes on new data. This situation reminds auto-associative property: too many neurons keep too much information about training set rather "remembering" than "learning" its characteristics. This is not enough to ensure good generalization that is needed.

On the other hand, when there are too few hidden neurons the network may never learn the relationships amongst the input data. Since there is no precise indicator how many neurons should be used in the construction of a network, it is a common practice to built a network with some initial number of units and when it trains poorly this number is either increased or decreased as required. Obtained solutions are usually task-dependant.

3.2 Activation Functions

Activation or transfer function of a neuron is a rule that defines how it reacts to data received through its inputs that all have certain weights.

Among the most frequently used activation functions are linear or semi-linear function, a hard limiting threshold function or a smoothly limiting threshold such as a sigmoid or a hyperbolic tangent. Due to their inherent properties, whether they are linear, continuous or differentiable, different activation functions perform with different efficiency in task-specific solutions.

For classification tasks antisymmetric sigmoid tangent hyperbolic function is the most popularly used activation function:

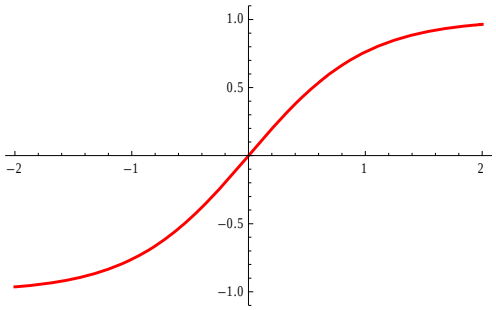

Fig. 1. Antisymmetric sigmoid tangent hyperbolic activation function

3.3 Learning Rules

In order to produce the desired set of output states whenever a set of inputs is presented to a neural network it has to be configured by setting the strengths of the interconnections and this step corresponds to the network learning procedure. Learning rules are roughly divided into three categories of supervised, unsupervised and reinforcement learning methods.

The term supervised indicates an external teacher who provides

information about the desired answer for each input sample. Thus in case of supervised learning the training data is specified in forms of pairs of input values and expected outputs. By comparing the expected outcomes with the ones actually obtained from the network the error function is calculated and its minimization leads to modification of connection weights in such a way as to obtain the output values closest to expected for each training sample and to the whole training set.

In unsupervised learning no answer is specified as expected of the neural network and it is left somewhat to itself to discover such self-organization which yields the same values at an output neuron for new samples as there are for the nearest sample of the training set.

Reinforcement learning relies on constant interaction between the network and its environment. The network has no indication what is expected of it but it can induce it by discovering which actions bring the highest reward even if this reward is not immediate but delayed. Basing on these rewards it performs such re-organization that is most advantageous in the long run [16].

The modification of weights associated with network interconnections can be performed either after each of the training samples or after finished iteration of the whole training set.

The important factor in this algorithm is the learning rate η whose value when too high can cause oscillations around the local minima of the error function and when too low results in slow convergence. This locality is considered the drawback of the backpropagation method but its universality is the advantage.

4. APPLICATIONS

Author identification analysis that was performed within research presented in this paper can be seen as the multistage process, as follows

the first step was selection of the training and testing examples - *texts to be studied,*
next stage was taken by the choice of textual descriptors to be analyzed - *the writerprints of the authors of previously selected texts,*
then followed the third phase of calculating characteristics for all descriptors that were later used for training of the neural network, *calculation,*
specification of the network with its architecture and learning method can be seen as the fourth step of the whole procedure, *neural network,*
the fifth consisted of the actual *training of the network*,
the sixth stage is *testing*,
and the final one corresponded to analysis of obtained results and coming up with some conclusions and possible indicators for improvement, *analysis of obtained results*.

This process is applied to different input data, with three committee machines of

neural networks, working together in a boosting by filtering method.

4.1 Texts Used

In research texts of two famous Bosnian writers, Ivo Andrić and M. Meša Selimović are used. Their novels provide the corpora which are wide enough to make sure that characteristic features found based on the training data can be treated as representative of other parts of the texts and this generalized knowledge can be used to classify the test data according to their respective authors.

Obviously literary texts can greatly vary in length; what is more, all stylistic features can be influenced not only by different timelines within which the text is written but also by its genre. The first of these issues is easily dealt with by dividing long texts, such as novels, into some number of smaller parts of approximately the same size.

Described approach gives additional advantage in classification tasks as even in case of some incorrect classification results of these parts the whole text can still be properly attributed to some author by based the final decision on the majority of outcomes instead of all individual decisions for all samples.

Whether the genre of a novel is reflected in lexical and syntactic characteristics of it is the question yet to be answered. If the influence is significant, then lexical and syntactic features cannot be used as the writer invariant as unreliable. On the other hand, this can be rectified by including within the training data set fragments of texts being representatives of not only one but several genres. In fact the more the better. For intended implementation of the classifier with Artificial Neural Networks, which efficiently deal with large amount of data, adding samples to the training set simply means better coverage of the input space that is important in continuous case.

Hence all together we have selected 1466 paragraphs coming from "na drini ćupria"[17] by Ivo Andrić, and "derviš i smirt"[18] by M. Meša Selimović each.

4.2 Feature Selection

Establishing features that work as effective discriminators of texts under study is one of critical issues in research on authorship analysis which are lexical. In this research five textual descriptors are used, numbers of characters, words, sentences, commas, and conjecture “and”, in Bosnian “i” in paragraphs. The descriptive statistics for these textual descriptors are as in Table 1 below:

Table 1. Paragraph averages and variances of the textual descriptors used in this research

	Ivo Andrić		M. Selimović	
Textual descript	Mean	Variance	Mean	Variance
Charact	367	131292	286	117193
Words	78.7	5979.8	62.1	5518.1
Sentence	4.33	15.694	4.60	26.7
Commas	6.45	47.543	7.5	107.8
“i”	5.35	35.506	2.36	11.4

As it is seen, there is statistical difference between the usage of textual descriptors, for instance, Ivo Andrić prefers longer paragraphs. In average Ivo Andrić 's paragraphs contain 79 words with variance 5080, while Meša Selimović's average is 62 with variance 5518. Our neural networks will capture this pattern during the training phase, and use this information to classify the paragraphs in the test data.

4.3 Architecture of artificial neural networks, Committee Machines

As the base topology of artificial neural network committee machines [5] with the feed-forward multilayer perceptron with sigmoid activation function trained by backpropagation algorithm is used.

In committee machines approach, a complex computational task is solved by dividing it into a number of computationally simple tasks and then combining the solutions to those tasks. In supervised learning, computational simplicity is achieved by distributing the learning task among a number of experts, which in turn divides the input space into a set of subspaces. The combination of experts is said to constitute a committee machine. Basically, it fuses knowledge acquired by experts to arrive at an overall decision that is supposedly superior to that attainable by anyone of them acting alone. The idea of a committee machine may be traced back to Nilsson [19] (1965); the network structure considered therein consisted of a layer of elementary perceptrons followed by a vote-taking perceptron in the second layer.

Committee machines are universal approximators. They may be classified into two major categories:

1. *Static structures.* In this class of committee machines, the responses of several predictors (experts) are combined by means of a mechanism that does not involve the input signal, hence the designation "static." This category includes the following methods:

• Ensemble averaging, where the outputs of different predictors are linearly combined to produce an overall output.

• Boosting, where a weak learning algorithm is converted into one that achieves arbitrarily high accuracy.

2. *Dynamic structures.* In this second class of committee machines, the input signal is directly involved in actuating the mechanism that integrates the outputs of the individual experts into an overall output, hence the designation "dynamic."

Boosting

Boosting is a method that belongs to the "static" class of committee machines. Boosting is quite different from ensemble averaging. In a committee machine based on ensemble averaging, all the experts in the machine are trained on the same data set; they may differ from each other in the choice of initial conditions used in network training. By contrast, in a boosting

machine the experts are trained on data sets with entirely different distributions; it is a general method that can be used to improve the performance of any learning algorithm.

Boosting' can be implemented in three fundamentally different ways:

1. Boosting by filtering. This approach involves filtering the training examples by different versions of a weak learning algorithm. It assumes the availability of a large (in theory, infinite) source of examples, with the examples being either discarded or kept during training. An advantage of this approach is that it allows for a small memory requirement compared to the other two approaches.

2. Boosting by subsampling. This second approach works with a training sample of fixed size. The examples are "resampled" according to a given probability distribution during training. The error is calculated with respect to the fixed training sample.

3. Boosting by reweighting. This third approach also works with a fixed training sample, but it assumes that the weak learning algorithm can receive "weighted" examples. The error is calculated with respect to the weighted examples.

In this paper *Boosting by filtering* is used. This algorithm is due to Schapire [20] (1990). The original idea of boosting described in Schapire (1990) is rooted in a distribution free or probably approximately correct (PAC) model of learning. To be more specific, the goal of the learning machine is to find a hypothesis or prediction rule with an error rate of at most ε, for arbitrarily small positive values of ε, and this should hold uniformly for all input distributions.

In boosting by filtering, the committee machine consists of three experts or subhypotheses. The algorithm used to train them is called a boosting algorithm. The three experts are arbitrarily labeled "first," "second," and "third." The three experts are individually trained as follows:

1. The first expert is trained on a set consisting of N, examples.

2. The trained first expert is used to filter another set of examples by proceeding in the following manner:

Flip a fair coin; this in effect simulates a random guess.
If the result is heads, pass new patterns through the first expert and discard correctly classified patterns until a pattern is misclassified. That misclassified pattern is added to the training set for the second expert.
If the result is tails, do the opposite. Specifically, pass new patterns through the first expert and discard incorrectly classified patterns until a pattern is classified correctly. That correctly classified pattern is added to the training set for the second expert.
Continue this process until a total of N, examples has been filtered by the first expert. This set of filtered examples

constitutes the training set for the second expert.

By following this coin flipping procedure, it is ensured that if the first expert is tested on the second set of examples, it would have an error rate of 1/2. In other words, the second set of N, examples available for training the second expert has a distribution entirely different from the first set of N, examples used to train the first expert. In this way. the second expert is forced to learn a distribution different from that learned by the first expert [21].

3. Once the second expert has been trained in the usual way, a third training set is formed for the third expert by proceeding in the following manner:

• Pass a new pattern through both the first and second experts. If the two experts agree in their decisions, discard that pattern. If, on the other hand, they disagree, the pattern is added to the training set for the third expert.

• Continue with this process until a total of N, examples has been filtered jointly by the first and second experts. This set of jointly filtered examples constitutes the training set for the third expert.

The third expert is then trained in the usual way, and the training of the entire committee machine is thereby completed.

Let N_2 denote the number of examples that must be filtered by the first expert to obtain the training set of N_1, examples for the second expert. Note that N_1 is fixed, and N_2 depends on the generalization error rate of the first expert. Let N_3 denote the number of examples that must be jointly filtered by the first and second experts to obtain the training set of N_1 examples for the third expert.

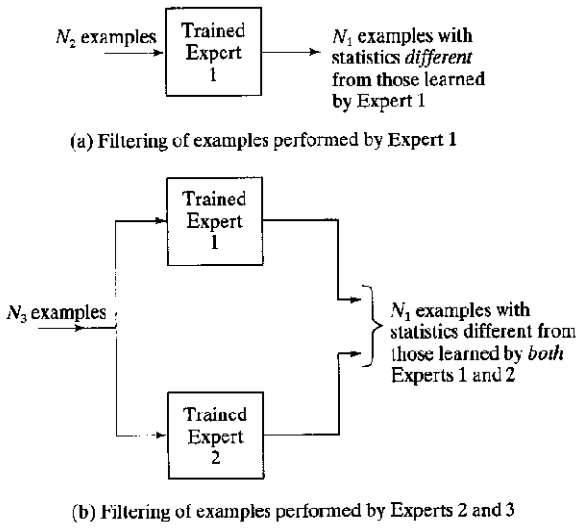

Fig. 2.The three-point filtering procedure

With N_1 examples also needed to train the first expert, the total size of data set needed to train the entire committee machine is $N = N_1 + N_2 + N_3$. However, the computational cost is based on $3N_1$ examples because N_1 is the number of examples actually used to train each of the three experts. We may therefore say that the boosting algorithm described herein is indeed "smart" in the sense that the committee machine requires a large set of examples for its operation, but only a subset of that data set is used to perform the actual training.

Another noteworthy point is that the filtering operation performed by the first expert and the joint filtering operation

performed by the first and second experts make the second and third experts, respectively, focus on "hard-to-learn" parts of the distribution.

In the theoretical derivation of the boosting algorithm originally presented in Schapire (1990), simple voting was used to evaluate the performance of the committee machine on test patterns not seen before. Specifically, a test pattern is presented to the committee machine. If the first and second experts in the committee machine agree in their respective decisions, that class label is used. Otherwise, the class label discovered by the third expert is used. However, in experimental work presented in Drucker et al.[22-23] (1993,1994), it has been determined that addition of the respective outputs of the three experts yields a better performance than voting. For example, in the optical character recognition (OCR) problem, the addition operation is performed simply by adding the "digit 0" outputs of the three experts, and likewise for the other nine digit outputs.

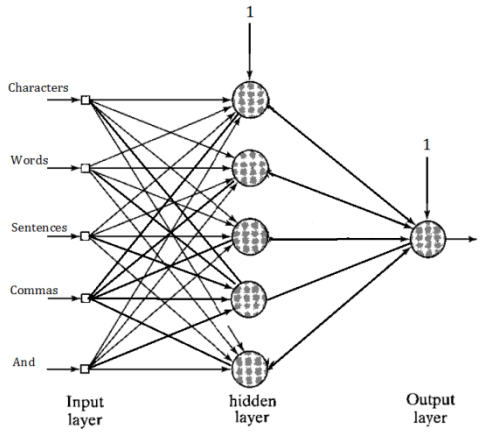

Fig. 3. Signal flow graph of each of the three expert machines

The number of inputs equaled the number of textual descriptors used, thus it is five. There is one hidden layer with five neurons within each of three neural networks in the committee machine for preserving generalization properties but achieving convergence during training with tolerance at most 0.14 for all training samples recognized properly.

For all structures of artificial neural networks, only one output is produced. Actually, it was possible to use a single output and by interpretation of its active state as one class and inactive output state the second class the task would have been solved as well, but with such approach the text is attributed to either one or another author and classification is binary. Algorithm results in a decision about attribution of paragraphs whose textual description entered as inputs.

5. RESULTS AND DISCUSSION

For validation purposes 200 samples are used from some other parts of the same works of both writers. As lexical descriptors, numbers of characters, words, sentences, commas, and conjecture “and”, in Bosnian “i” in paragraphs are chosen.

Set 1 of data consists of lexical descriptors from 200 paragraphs chosen from both novels of each author. N_1=400 is the number of data to train the first machine of the committee which has two input terminals, thirteen hidden neurons in one hidden layer. The results of classification performed at the end of training by this network machine are given in the Table 2 below.

Table 2. Number of correct classifications of paragraphs in the training data at the end of the training period of the first committee machine.

	Data Number	Correct Classification	%Corrct Classification
Ivo	200	158	79
Meša	200	130	65
Total	400	288	72

Then another set of $2N_1$ descriptors sent to the first machine. This data are distinguished in two classes: C_1, the class of correctly classified data, C_2 , the class of incorrectly classified data. Then a coin is tossed. If heads come up, a data is taken from the class C_1, otherwise, a data is taken from the class C_2 is picked up to form a training set of N_1 data for the second machine of the committee.

The results of classification performed at the end of training by this second network machine are given in the Table 3 below.

Table 3. Number of correct classifications of paragraphs in the training data at the end of the training period of the second committee machine.

	Data	Correct Classification	% Correct Classification
Ivo	200	112	56
Meša	200	150	75
Total	400	262	65.5

Next, samples which are not used for training before are sent to the first and second committee machines. If the two machines agree on the classification this data is thrown out. Otherwise it is kept to form the N_1 data to train the third machine of the committee.

Table 4. Number of correct classifications of paragraphs in the training data at the end of the training period of the third committee machine.

	Data Number	Correct Classification	% Correct Classification
Ivo	200	110	55
Meša	200	150	75
Total	400	260	65

Although personal success rates are low, seemingly the first machine is an expert for Ivo, and the third machine is a Meša expert compared to other machines in the committee.

Finally the test set sent to three experts. Their personal performances in classifying the test data is given in Table 4.

Table 5 Personal and overall performances of three experts in classifying the test data

	Data Nu	I	II	III	% Av
Ivo	200	152	134	154	73.00
Meša	200	131	153	148	74.50
Total	400	283	287	302	73.75

Combining Results

To combine the results, we ensemble decisions of each machine simply taking the average of the decisions of the three experts. Committee performance in

classifying the test data is given at the last column of Table 5.

As it is seen from Table 5, the committee success is satisfactory, 73% of the paragraphs in the test data authored by Ivo Andrić are correctly identified. A higher percentage of 74.5% of the paragraphs authored by Meša Selimović is identified correctly. Overall correct classification probability is high enough, 73.75%. There is 26.25% of misclassification. 54 out of 200 paragraphs of Ivo Andrić, and 51 out of 200 paragraphs of Meša Selimović are identified incorrectly.

6. CONCLUSIONS

The research described in this paper concerning author identification analysis shows beyond doubt how efficient a tool Artificial Neural Networks can be when applied in classification tasks. Yet conclusions as to the choice of textual descriptors used as features for recognition process, based only on results presented in the previous section and leading to some arbitrary statement that syntactic attributes are more effective in authorship attribution, would be much too hasty and premature. Undeniably true in the studied example, it would have to be verified against much wider corpora as for other writers other features could give better results.

Thus a series of future experiments should include application of the presented here artificial neural networks -based methodology to wider range of authors, definition of new sets of textual descriptors, and test for other types and structures of neural networks, and search the possibility of inheritance through translation into other languages.

REFERENCES

[1] N. McCombe, Methods Of Author Identification, Final Year Project, May 2002.

[2] S. Doan, S. Horiguchi, “An efficient feature selection using multi-criteria in text categorization for naive Bayes classifier”, WSEAS Transactions on Information Science & Applications, vol. 2, no. 2, pp. 98–103, 2005

[3] T. Taş, A. K. Görür, Author Identification for Turkish Texts, Çankaya Üniversitesi Fen-Edebiyat Fakültesi, Journal of Arts and Sciences No: 7, May 2007

[4] S. Gazzah, and N. Ben Amara, Neural Networks and Support Vector Machines Classifiers for Writer Identification Using Arabic Script, The International Arab Journal of Information Technology, Vol. 5, No. 1, January 2008.

[5] S. Haykin, Neural Networks A Comprehensive Foundation, Second Edition, Prentice-Hall, Inc.

Simon & Schuster,A Viacom Company Upper Saddle River, New Jersey 07458, 1999.

[6] U. Stańczyk, K. A. Cyran, Machine learning approach to authorship attribution of literary texts, International Journal Of Applied

Mathematics And Informatics, Issue 4, Volume 1, 2007, pp. 151-158.

[7] R.A.J. Matthews and T.V.N. "Merriam, Distinguishing literary styles using neural networks", in E. Fiesler and R. Beale, eds., Handbook of neural computation, OUP, pp. G8.1.1–6, 1997.

[8] R.D. Peng and H. Hengartner, "Quantitative analysis of literary styles, The American Statistician", vol. 56, no. 3, pp. 15–38. 2007.

[9] J. M. Zurada, Introduction to artificial neural systems, West Publishing Company, 1992.

[10] M. Zi¸E. Swi¸ and J. Atek, Two-stage Writer Identification Using Complex Neural Network System,

[11] M. Rosenblatt, M.,1970. "Density estimates and Markov sequences." in M. Puri, ed., Nonparametric Techniques in Statistical Inference, pp.199-213, London: Cambridge University Press.

[12] W. S. Mcculloch, and W.Pill's. (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity." Bulletin of Mathematical Biophysics, 5:115-133. Reprinted in Anderson& Rosenfeld [1988], pp. 18-28.

[13] D. O. Hebb, (1949). The Organization of Behavior. New York: John Wiley & Sons. Introduction

and Chapter 4 reprinted in Anderson & Rosenfeld [1988], pp. 45-56.

[14] Rosenblatt, E, 1958."The Perceptron:A probabilistic model far information storage and organization in the brain," Psychological Review, vol. 65, pp. 386-408.

[15] M. L. Minsky, and S. A. Papert, (1988). Perceptrons, Expanded Edition. Cambridge, MA:

MIT Press. Original edition, 1969.

[16] H. Tang, K. C. Tan, and Z. Yi, Neural Networks: Computational Models and Applications, Springer-Verlag Berlin Heidelberg 2007.

[17] I. Andrić, Na Drini ćuprija

[18] M. M. Selimović, Derviš i smrt,1966.

[19] N. J. Nilsson, Learning Machines: Foundations of Trainable Pattern-Classifying Systems, New York: Mcffraw-Hill, 1965.

[20] R. E. Schapire, R.E, "The strength of weak learnability," Machine Learning, vol. 5, pp.197-227, 1990.

[21] R. E. Schapire, 1997. "Using output codes to boost multiclass learning problems," Machine Learning: Proceedings of the Fourteenth International Conference, Nashville, TN.

[22] H. Drucker, C. Cortes, L.D. Jackel, and Y. LeCun, 1994. "Boosting and other ensemble methods." Neural Computation, vol. 6, pp.1289-1301

[23]H. Drucker, R.E. Schapire, and P. Simard, 1993."Improving performance in neural networks using a boosting algorithm," Advances in Neural Information Processing Systems, vol. 5, pp. 42--49, Cambridge, MA: MIT Press.