FEED-FORWARD AND SELF-ORGANIZING NEURAL NETWORKS FOR TEXT DOCUMENT RETRIEVAL

Igor MOKRIŠ, Lenka SKOVAJSOVÁ
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
mokris@valm.sk, skovajsova@valm.sk

ABSTRACT

The aim of this paper is to survey the feed-forward and self-organizing neural networks for the text document retrieval models, which retrieve text documents in a natural language. These models come from linguistic and conceptual approach of the text document analysis, where problems of document representation and document database creation are being solved. The proposed structure of the feed-forward and self-organizing neural network models solve the problem of the document retrieval by a user's query which is transformed into the set of keywords. However, learning algorithm and neural network invariance, which comes from utilization of the chosen neural networks, enable the decrease of computational complexity for the language analysis of text document retrieval process.

Keywords: *text document retrieval, queries, keywords, feed-forward neural networks, self-organizing neural networks*

1. INTRODUCTION

Expansion of electronic documents processing started mainly by the introduction of internet and electronic libraries. Simultaneously to the great expansion of documents there appeared systems for the information retrieval from text documents in a natural language.

Information retrieval system enables to obtain needed information from the text document set by means of a user query [2]. User sends a query to the information retrieval system; the system creates an inner representation of that query and transforms it into keyword set and then retrieves the relevant documents from the document collection. Association or context of query and document is known as document relevance.

Information retrieval system is represented by a query, a document and an information retrieval model. The query can be represented by a pattern, keyword vector or by a structured query. Documents can be represented by an index, semantic network, ontology, etc. [2], in a set of document collection.

The manner in which the documents can be retrieved from the text document collection is described by the text document retrieval model. In the beginning there were the Boolean models, vector and probabilistic models; later also the neural network models were used [2, 6, 19].

For the evaluation of the information retrieval models different parameters were used. From these parameters the most important is precision, recall, and relevance feedback. Precision and recall is based on a relevant document set evaluation in relation to the retrieved document set. Relevance feedback enables to modify given query and then make the information retrieval more qualified. The advantage of this approach is to make the document retrieval more transparent but the disadvantage is that at the beginning of this process the user needs always to give a query to which the user must find relevant documents alone.

Systems that learn from relevance feedback are trying to modify the representation of the query, retrieval function or document representation [7, 8, 23]. Methods based on the modification of query representation enable to obtain more relevant information for new query by means of the older query. Methods based on the retrieval function modification enable to obtain new retrieval function to improve the retrieval process. The principle of the methods based on the document representation modification is to obtain better representation of the document collection. This methods use three main approaches [23]. First of them use the Brauen modification of document representation. A disadvantage of this method is the control deficiency over the learning process. This disadvantage is eliminated by the second approach which was solved by Bodoff stress function. The third approach uses probabilistic model. The principle of this approach is to learn the relation between each query and specific document from relevance feedback.

There are two basic approaches in information retrieval, i.e. ad-hoc information retrieval and filtering information retrieval [2].

For description of the information retrieval systems were proposed a lot of models that enabled formalization of information retrieval processes. In the following text the paper deals with description of the models for information retrieval systems to obtain text documents in natural language.

2. INFORMATION RETRIEVAL MODELS

A lot of models for text document retrieval were proposed. Most often used is the Boolean model, the vector space model, the latent semantic model, the probabilistic model and also neural network models.

Historically oldest is the Boolean model. It is based on the set theory and Boolean algebra [1, 7]. Documents are stored in the database by inverted index. The advantage of the Boolean model is its simplicity for document retrieval description; the disadvantage is its high complexity for the structured query definition. There are other disadvantages of this model, e.g. too many or too few documents in response to a query, the unability to sort documents by their relevance or not taking term frequency into account. Some of these problems solve the extended Boolean model [2, 7].

Probably the most used model for the information retrieval is the vector space model [27]. Each document in the document collection is defined by its keyword vector and each element of document representation in the vector space model is expressed by the frequency of each keyword in the document. This way the vector space matrix is determined.

The latent semantic model comes from the vector space model where for the representation and reduction of the vector space matrix a singular value decomposition is used (SVD) [2, 9, 12]. It has two main advantages in comparison with vector space model. These are document space dimension reduction and catching similarity between documents that have different keywords. SVD algorithm can be used in the document retrieval model for principal component analysis, which expresses the document relevance. Similarly, the independent component analysis can be used for extraction of statistically independent documents [4, 12].

Probabilistic model uses Bayesian conditional probabilities for association of the query and documents in a document set. The most significant probabilistic models are the binary probabilistic models, probabilistic relevance feedback models, Bayesian networks, inference network models and belief nets [2].

Recently, neural networks were applied in information retrieval models [2, 6, 19]. The advantage of this approach is the fact that in some cases it is difficult to find relations between quantities of the information processes. Therefore, these relations can be substituted by learning processes of the neural networks. Similarly, parallel structure and invariance of neural networks enable to speed up and simplify algorithms and processes of information retrieval.

This paper is focused on the description of models for text document retrieval in a natural language by the feed-forward and self-organizing neural networks and results which were obtained by them.

3. FEED-FORWARD NEURAL NETWORKS FOR INFORMATION RETRIEVAL MODELS

This type of neural networks is very often used for information retrieval models [2, 3, 18, 19]. The advantage of these neural networks is the simplicity of model description.

3.1. Information retrieval by COSIMIR system

System COSIMIR (**CO**gnitive **SIM**ilarity learning in **I**nformation **R**etrieval) [18] is proposed by a feed-forward neural network of back-propagation type and determines relevance between the query and the document (Fig. 1). The input of a neural network is represented with a query and a document; a query and a document are represented by a keyword vector. The output of a neural network determines the relevance between this query and document.

A similar system to COSIMIR is the system described in [3]. The feed-forward neural network of back-propagation type is also used there. The input layer consists of a binary representation of a word vector and the documents are represented by tf-idf weight scheme. The middle layer is used for the representation of a word and a document and the output layer determines how this word relates with document.

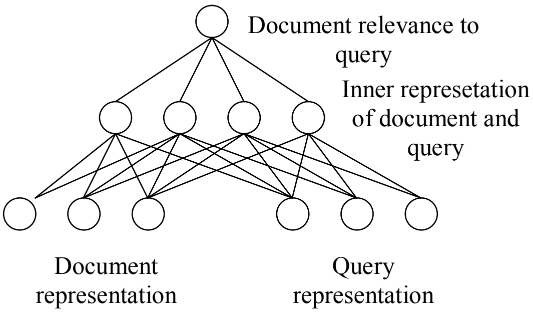
Document relevance to query
Inner represetation of document and query
Document representation
Query representation

Fig. 1 System COSIMIR

3.2. Information retrieval by feed – forward neural network with spreading activation function

Very often used representation of document set by keyword vector in Information retrieval models is the vector space model [2]. The vector space model is described by relative frequency matrix *F* of keywords in document set. The vector space model is created by rows of columns of documents d_1, ... , d_p which contain the keywords k_1, … , k_L and by relation

$$F(L\times P)=\begin{pmatrix}k_1\\k_2\\\cdots\\k_L\end{pmatrix}=\begin{array}{c}\begin{array}{cccc}\mathrm{d}_1&\mathrm{d}_2&&\mathrm{d}_P\end{array}\\\begin{pmatrix}f_{11}&f_{12}&\cdots&f_{1P}\\f_{21}&f_{22}&\cdots&f_{2P}\\\cdots&\cdots&\cdots&\cdots\\f_{L1}&f_{L2}&\cdots&f_{LP}\end{pmatrix}\end{array}\tag{1}$$

where d_j is j-th document, P is the number of documents, k_i is i-th keyword, L is the number of keywords, f_{ij} is relative frequency of the i-th keyword in the j-th document called also keyword weight.

The query comes to this model by means of corresponding keyword set and the result is the document relevance to this query. Documents with greater relevance as is the given threshold are arranged by their relevance and are turning to the user as a result.

For representation of the vector space model was used the neural network with spreading activation function [2, 3, 5, 27] which can be represented as a feed -forward neural network with linear activation function (purelin)

$$f(net_{dj})=net_{dj}\tag{2}$$

It has two layers, the input and output layer (Fig. 2). Each neuron in the input layer represents one keyword; the set of keywords represents the query. Each output neuron represents associated document or particular document relevance to the given query.

This neural network is not trained. Its weight matrix *W* is acquired by assigning of vector space matrix F

$$W=F\tag{3}$$

and the document vector $D = (d_1, d_2, ..., d_n)$ is calculated by

$$d_j = w_{1j} k_1 + w_{2,j} k_2 + ... + w_{mj} k_m \qquad (4)$$

Large disadvantage of this approach is high dimension of vector space matrix for large number of documents.

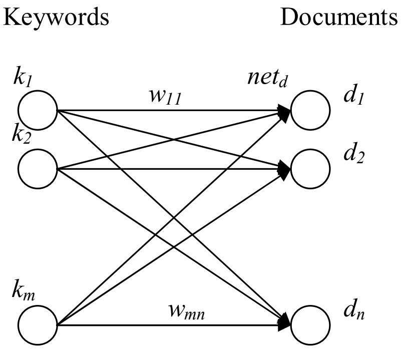

Fig. 2 Neural network with spreading activation function

3.3. Document space dimension reduction based on LSI model

For large document collections the dimension of the vector space matrix *F* is high, what causes problems in text document set representation and high computing complexity in information retrieval. The dimension reduction of the vector space matrix solves Latent Semantic Indexing (LSI) [6, 9, 26]. LSI uses singular value decomposition (SVD) of a vector space matrix F to divide it on three matrices as can be seen

$$F = U . S . V \qquad (5)$$

where $F = F(i \times j)$ is VSM matrix, $U = U(L \times M)$ is matrix of left eigenvectors, $S = S(M \times M)$ is diagonal matrix of positive singular values, and $V = V(M \times P)$ is the matrix of right eigenvectors.

Dimension reduction of matrix F is performed by substitution of $s_k \approx 0$ by $s_k=0$

$$s_k \approx 0 \rightarrow s_k=0 \qquad (6)$$

and matrix of singular values S will be approximated by matrix of approximated singular values $S \approx S_R$, $R<M$ and formula (5) will be substituted by formula

$$F_R = U . S_R . V^T \qquad (7)$$

After the dimension reduction, reduced matrix F_R has fewer elements as original vector space matrix *F*.

Decomposition of VSM matrix K can be performed by auto-associative neural network with three layers (Fig.3). Input and output layer represents documents expressed by keyword set in document set, hidden layer represents reduced document space where each hidden neuron represents its dimension. *W1* and *W2* is weight vector of hidden and output layer ind c_i is hidden neuron for i – reduced dimension of document set.

Input and output layer represents one document which is represented by a keyword vector

$$D_i = (k_{1i}, ..., k_{Mi}) \quad \text{for } i = 1, ..., N \qquad (8)$$

Each input and output neuron represents the same keyword k_{ki} from the keyword vector k_k of document d_k. In the hidden layer of the neural network, the number of neurons is equal to the number *R* of approximated singular values of the reduced LSI matrix S_R. During training phase the neural network uses the document vectors of VSM matrix, that are represented by keyword set and this way the hidden neurons are setting by back-propagation learning algorithm.

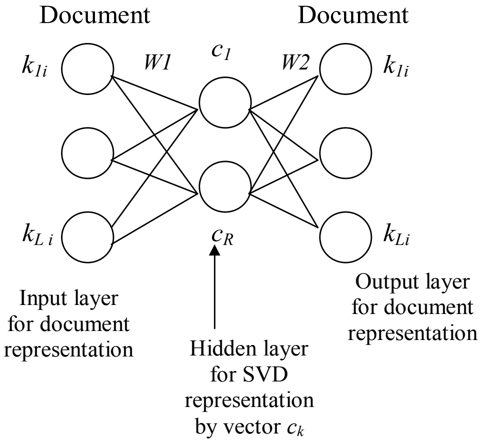
Document
Document
k_{1i}
$W1$
c_1
$W2$
k_{1i}
$k_{L\,i}$
c_R
k_{Li}
Input layer for document representation
Hidden layer for SVD representation by vector c_k
Output layer for document representation

Fig. 3 Auto-associative neural network for LSI

In the phase of testing the input documents come on the input of the network. The result is the vector c_k in the reduced document space

$$c_i = d_i WI \qquad (9)$$

which solves the SVD computation problem.

3.4. Document space dimension reduction based on PCA method

The most common method of the document space dimension reduction, which represents analyzed documents, is the Principal Component Analysis (PCA) [9, 12]. The principal components of the projected document space are not correlated and have maximal dispersion. For the PCA representation by the feed-forward neural network, the associative memory with Generalized Hebbian Learning Algorithm (GHA) can be used. This neural network on the PCA base was used for reduction keywords on principal components.

3.5. Feed-forward neural network for relevance feedback model

For the relevance feedback model the three-layer feed – forward neural network was used [9]. Each input neuron represents a query and each output neuron represents a document. In the learning phase the user query was used as an input and the network was trained on the relevant output documents represented by a set of keywords. In the life phase the input was a query and the output was the first m - mostly activated new queries which are connected with given query (Fig. 5).

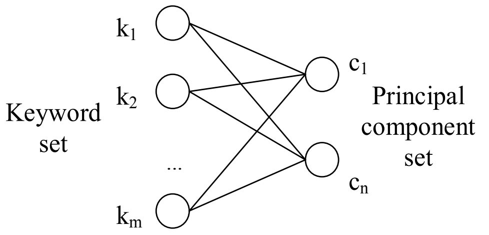

Fig. 4 Hebbian network

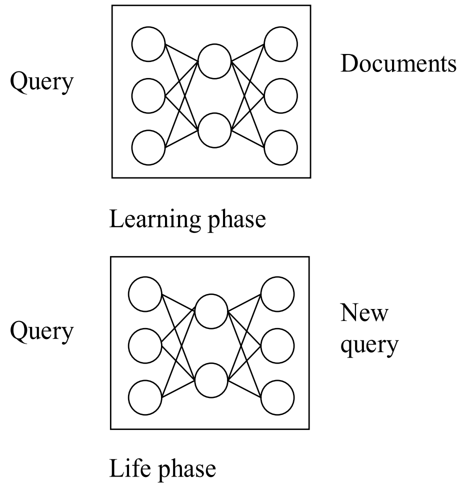

Fig. 5 Relevance feedback neural network

4. INFORMATION RETRIEVAL MODELS BY KOHONEN SELF-ORGANIZING MAPS

Kohonen self-organizing maps (SOM) [4, 11, 13 14, 16, 21, 22, 25] use competition based learning rule. They have two layers and the output layer is arranged into two-dimensional map with full connection.

4.1. Information retrieval by SOM based system WEBSOM

The WEBSOM system uses Kohonen SOM for document mapping [14, 16, 11]. As an input to SOM can be presented a vector of concepts, where concept represents reduced dimension of document set, acquired by [11, 17] as:

1. random mapping method,
2. latent semantic indexing,
3. independent component analysis,
4. word category map.

This concept has p dimensions, where p<<n. In this paper the keyword representation vector of documents is used.

Let $k_i = \{k_{i1},...,k_{in}\} \in R^n$ be an input keyword vector of document d_i and each neuron in the output layer represents a document $d_j(t) \in R^n$, i.e., output layer represents all set of documents. To one output neuron can be assigned more documents with similar keywords or no document depending on its position. The SOM learning algorithm has two steps:

1. input vector of keywords k_i of document d_i is compared with a weight vectors W of all output neurons that represent the whole set of documents D. The most similar neuron in output layer is the winner d_w,
2. winner d_w is found by re-calculation of weights in the learning process their neighborhood.

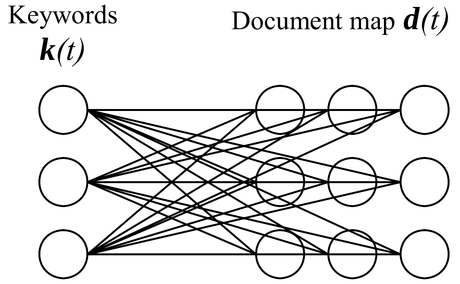

Fig. 6 Kohonen self-organizing map

For information retrieval in the large document collections the WEBSOM system was proposed in [13]. The development of WEBSOM system can be divided into four main steps [13]:

1. document encoding,
2. construction of large map,
3. construction of user interface,
4. alternative operations of user interface.

After the training of this neural network the relevant documents are placed near to each other. Information retrieval is enabled by visualization, where the user can see, what document is where on the SOM map. Keywords can be reduced to the lower number of concepts by Latent Semantic Indexing and the document retrieval task is simplified.

Independent component analysis method was compared with SOM in [4]. For small document collection have both methods similar results.

4.2. Information retrieval in the large collection of documents by SOM based system WEBSOM2

The WEBSOM system is too slow for large collections of data, so the WEBSOM2 system was developed which uses batch training algorithm so less computation in the training phase is made what speeds up the training process.

The experiments were made on 6 840 568 collection of abstracts in the electronic form in English language [13]. The headers were separated from the text at first. Then the non-text information was discarded. The dictionary contained 733 179 words. Then the stop-words and the words occurring less then the 50 times were removed. At the end more than 122 524 abstracts which contained less than 50 words were removed. For the dimension reduction the random projection method was used. This method shows how many times occurs given keyword in the text. The words were weighted by using the Shanon entropy. From these data the SOM was created and was enlarged to be able to represent large collections of data. The precision of retrieval acquired by this experiment was 64%.

4.3. Document classification by Kohonen self – organizing neural networks

In order to make the information retrieval more prime by the SOM information retrieval three new approaches were proposed [10]. First of them is the Extended Significant Vector Model (ESVM). ESVM comes out from the matrix of words and from the matched classes and from the vector space matrix.

The second approach, the Hypernym Significance Vector Model (HSVM), solves the synonymy and polysemy problem by ontologies. The approach used here solves next problem: If it is not clear to which category the document belongs, then there are two different ontologies, which contain that word. There are the neighbors searched and the ontology that contains more words from the neighborhood is the right ontology.

The third approach, the Hybrid Vector Space Model (HyM), combines the algorithm HSVM and classical SOM algorithm and improves both the algorithms. It uses a predefined parameter $\gamma \in (0,1)$, that influences the rate of supervised learning algorithm.

Experiments were made on the RCV1 Reuters corpus [10]. There are 806 791 news articles. Each document is saved in XML format and has three different codes: industry code, regional code and theme code. Experiments relate to eight different topics that are reclassified to 40 different topics. In these topics are classified 10000 articles. Measures of these approaches are twofold: quantitative and qualitative. Quantitative measures are internal as a quantization error, or external, for example classification accuracy. The quantization error has the following form

$$QE = \sum_{i=1}^{N} \left\| k_i - w_i \right\| \tag{10}$$

where w_{i} is the weight vector of the winner neuron, k_i is an input keyword and N is the total number of input vectors. From the quantization error can be counted the Average Quantization Error (AQE) and Mean Quantization Error (MQE).

Let U be the total number of output neurons. Then AQE is defined on the base

$$AQE = \frac{QE}{U} \tag{11}$$

Let N be the total count of input neurons. Then MQE is defined as

$$MQE = \frac{QE}{N} \tag{12}$$

The classification accuracy is counted as a relation of well classified articles to all input articles.

The two methods were compared, ESVM and SOM-based Vector Space Model. ESVM showed better distinguishable results. It was also shown that connection of the neuron model with WorldNet ontology gives better results as a neural model without using ontology.

4.4. SOM with LVQ learning algorithm for document classification and retrieval

When it is possible to assign the documents to predefined classes, the LVQ algorithm can be used for document classification and retrieval.

LVQ learning algorithm is the variant of the SOM learning which uses the supervised learning [15, 20]. It is a learning method based on competitive learning that enables to define a group of classes in the space of the input data by the reinforcement learning.

To each class **x** the keyword vector k_i is assigned. Algorithm chooses the input vector x_i and balances it with each weight vector w_k by using the Euclidean distance || $k_i - w_k$ || for all iterations. The winner will be the weight vector w_c the most near to k_i

$$\left\| k_i - w_i \right\| = \min_x \left\{ \left\| k_i - w_x \right\| \right\} \tag{13}$$

The classes compete in order to find the most suitable class to input keyword vector. Only the winner class modifies its weights by using reinforcement learning that can be positive or negative, depending if it is the right class or not. So when the class is near the keyword vector increases its weights. On the other side, if the class of winner is different of the keyword vector class the weights far from the winner decrease.

Let $k_i(t)$ be the input vector in time t and $w_x(t)$ be the weight vector for class x in time t. The following definition determines the basic learning process for LVQ algorithm

$$w_c(t+1) = w_c(t) + s.\alpha(t)[k_i(t) - w_c(t)] \tag{14}$$

where s = 0 if $\mathrm{k} \neq \mathrm{c}$ and $w_c(t)$ belongs to the same class and s = -1, otherwise. And for the learning rate is $\alpha(t)$, and $0 < \alpha(t) < 1$ is recommended.

The experiments with proposed method were made on the collection of 1073 financial and news articles [15]. The LVQ algorithm was compared with SOM algorithm and the slight improvement in the document classification by LVQ method against the SOM method was detected.

4.5. Document set representation by Growing Hierarchical Self Organizing Maps

Growing Hierarchical Self Organizing Map (GHSOM) system helps to create growing SOM [21, 22]. These are used when an unknown number of documents is in the collection because these networks can adapt their structure and so their size. The whole process of neural networks creation starts with simple SOM of the dimension 2x2 neurons. Each unit in this map can create a new SOM. It is possible to repeat this approach for the second layer, the third layer and also for the higher layers. Such a network can be applied in the system, where it is not known how much neurons will be needed.

The first point of the growth of the network is created by a deviation from the input data in the first layer. To this neuron is assigned the weight vector $d_0 = [w_{01},...,w_{0n}]^T$, where w_{ij} is j-th weight of i-th neuron. Input data deviation (mqe) is counted on the base

$$mqe_0 = \frac{1}{n}\|d_0 - k\| \tag{15}$$

Here *k* represents input vector of keywords and *n* is the number of input patterns.

After the evaluation of mqe_0 is the training of the neural network transferred on the second layer over the neuron d_0. On this level the new SOM is created, which has a minimal number of units, (for example 2x2). To each neuron of this network a random value w_i is assigned. The weight vectors have then the same dimension as the input samples. The learning process of SOM continues as usual. The learning rate α decreases with learning time.

For the adaptation of created SOM in the second layer, the MQE is computed

$$MQE_m = \frac{1}{u}\sum_i mqe_i \tag{16}$$

where u is the number of neurons in the SOM and mqe_i is counted analogically as mqe_0 (13).

The base of GHSOM is that each layer is used for some part of deviation from the input signal. This is made by an addition of units to the given layer of GHSOM. SOM can grow in each layer until the deviation of the preceding layer is reduced to the value τ_m or less. The lower the parameter τ_m is, the larger the new SOM is. For that reason until it is true that $MQE_m \geq \tau_m mqe_0$ for the first layer of map m, a new row or column is given to this map. This addition is made in the neighborhood of the neuron e with the largest value of mqe_e after the λ training iterations. After this addition the α value is set for its starting value and the training continues by the standard training process of the SOM. If $MQE_m \leq \tau_m mqe_0$, the growing process of SOM is stopped and the new layer is created.

5. CONCLUSION

The aim of this paper was to survey the feed forward and self organizing neural networks for text document retrieval system modeling. On the base of published results it can be said that the most frequently used model for document representation is the vector space model. Other models like boolean model and probabilistic models are more – less used only occasionally. Boolean model is very simple and is less accurate and then less used. Probabilistic models have more theoretical signification. Also when the VSM model is most suitable for document representation, it disadvantage is its high dimension of its matrix for large document collection. Its neural network representation leads to great structural complexity given by the large number of input and output neurons and synapses.

In the text document retrieval of systems on the VSM model base, feed – forward neural networks were used, that obviously reflect the information retrieval system structure. Their structure is relatively simple and their learning process is in the long term used. For this reason is their application simple and easily verified. They are suitable for text document retrieval and classification on the keyword base for predefined document set structures. They are not suitable for large document collections.

Document space dimension reduction based on VSM model can be realized by the LSI model on the base of SVD decomposition but SVD decomposition is computationally complex. The computation complexity problem can solve the auto-associative neural networks or PCA-based neural networks.

In this area have its place also self-organizing neural networks that enable to realize the document classification on the supervised learning principles and enable to realize document clustering on the base of unsupervised learning also for large document collections. Their next advantage is the document set visualization in the Kohonen layer.

In the case of growing SOM these neural networks enable to adapt the SOM neural network structure by the structure of modeled documents and this way can be suitable to realize the hierarchical models of the document collection. This principle can be used for document space dimension reduction and so it allows to reduce computational complexity of text document retrieval model.

It is important to mention, that the keywords must be preprocessed at first. It means, that by user query which consist of the set of relevant meaningful words the keywords can be obtained. This stemming problem of relevant meaningful words by Porters algorithm [24] can be solved. In this case the neural networks thank invariability property can help solve the problem of keywords´ finding.

The advantage of above mentioned neural networks is simplicity of information retrieval systems modeling. The disadvantage of these neural networks is that nowadays state of theory does not solve the problem of keyword context that expresses the semantic character of text documents processing. Their application comes out from the user query syntax evaluation. On the base of survey in given area we can say that neural networks are promising approach for text document retrieval in natural language.

REFERENCES

[1] Amari S.I., Cichocki A.: Yang H.H.: A New Learning Algorithm for Blind Source Separation. Advances in Neural Information Processing Systems, 1996, pp.757-763.

[2] Baeza-Yates, Ribeiro-Neto B.: Modern Information Retrieval, Addison-Wesley, ISBN 0-201-39829-X, 1999.

[3] Bengio S., Keller M.: A Neural Network for Text Representation. International Conference on Artificial Neural Networks ICANN, 2005, pp. 667-672.

[4] Bingham E., Kuusisto J., Lagus K.: ICA and SOM in Text Document Analysis. Proceedings of 25th ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp. 361-362.

[5] Chau M., Chen H.: Incorporating Web Analysis Into Neural Networks: An Example in Hopfield Net Searching. Systems, Man, and Cybernetics, 2007, pp. 352-358

[6] Chen P. H.: Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning and Genetic Algorithms. JASIS 46(3), 1995, pp. 194-216.

[7] Choi J., Kim M., Raghavan V. V.: Adaptive Feedback Methods in an Extended Boolean Model. Proceedigs of ACM SIGIR Workshop on Mathematical Formal Methods in Information Retrieval, New Orleans, L.A, 2001.

[8] Crestani F.: Comparing Neural and Probabilistic Relevance Feedback in an Interactive Information Retrieval System. In Proceedings of the IEEE International Conference on Neural Networks, Orlando, Florida, USA, June 1994, pp. 3426–2430.

[9] Delichère M., Memmi D.: Neural Dimensionality Reduction for Document Processing, ESANN'2002 Proceedings of Europ. Symposium on Artificial Neural Networks, Bruges (Belgium), ISBN 2-930307-02-1, 2002, pp. 211-216.

[10] Hung C., Wermter S.: Neural Network-based Document Clustering using WordNet Ontologies. International Journal of Hybrid Intelligent Systems, Vol. 1, 2004, pp. 127-142.

[11] Kaski S. et. al.: WEBSOM Self Organizing Maps of Document Collections. Neurocomputing, 1998, pp. 101-117

[12] Kim Y. H., Zhang B.: Document Indexing using Independent Component Analysis and Signal Separation (ICA2001), San Diego, California, 2001, pp.557-562.

[13] Kohonen T. et. al.: Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, Vol. 11, No. 3, 2000, pp. 574-585.

[14] Kohonen T.: Self Organizing Maps, ISBN 3-50-67921-9, Springer, Berlin, 2001.

[15] Kuflik T. et. a.: Supervised Learning for Automatic Classification of Documents using Self Organizing Maps. DELOS Workshop , 2000, pp.

[16] Lagus K., Kaski S.,Kohonen T.: Mining Massive Document Collection by the WEBSOM method. Elsevier, 2004, pp. 135-156

[17] Lin J.,Gunopulos D.: Dimensionality Reduction by Random Projection and Latent Semantic Indexing. Proceedings of The Text Mining Workshop, SIAM 2003.

[18] Mandl, T.: Das COSIMIR Modell: Information Retrieval mit dem Backpropagation Algorithmus. ELVIRA-Arbeitsbericht 10, IZ, Bonn, 1999, pp.54-60.

[19] Mandl T.: Tolerant and Adaptive Information Retrieval with Neural Networks. Global Dialog, Science and Technology, Thinking the Future. EXPO 2000, Hannover, 2000, pp. 280-289.

[20] Martínez-Santiago F. et. al.: Using Neural Networks for Multiword Recognition in IR. Proceedings of Conference of International Society of Knowledge Organization (ISKO-02), Granada, Spain, 2002, pp.559-564.

[21] Merkl D., Rauber A.: Document Classification with Unsupervised Artificial Neural Networks. Soft Computing in Information Retrieval: Techniques and Applications. Heidelberg, Germany, Physica-Verlag, 2000, vol. 50, pp. 102-121.

[22] Merkl D., Dittenbach M., Rauber A.: Uncovering Hierarchical Structure in Data Using the Growing Hierarchical Self-Organizing Map. Neurocomputing, 2002, pp. 199-216.

[23] Piwowarski B.: Learning in Information Retrieval: a Probabilistic Differential Approach. Proceedings of the BCS-IRSG, 22nd Annual Colloquium on Information Retrieval Research, Sidney Sussex College, Cambridge, England, 2000.

[24] Porter, M. F.: An Algorithm for Suffix Tripping, Readings in Information Retrieval, Morgan Kaufmann Publishers, inc. 1997, pp. 313-316.

[25] Self Organizing Maps in Natural Language Processing. http://www.cis.hut.fi/~tho/thesis/

[26] Syu I., Lang S. D., Deo N.: Incorporating Latent Semantic Indexing into Neural Network Model for Information Retrieval. CIKM, 1996, pp. 145-153.

[27] Vector Space Model (VSM). http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node5.html

ACKNOWLEDGEMENT

This work was supported by the Grant Agency of Slovak Republic VEGA grant No. 2/7098/27.

Received May 28, 2007, accepted April 29, 2008

BIOGRAPHIES

Igor Mokriš (prof., Ing., PhD.) is scientist in Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia. His current research interest is oriented into the knowledge systems and neural networks.

Lenka Skovajsová (MSc.) was born in 1979. She received MSc degree in telecommunications from the Military Academy at Liptovský Mikuláš in 2003. Since 2004 she is studying her PhD. study in the Institute of Informatics, Slovak Academy of Sciences Bratislava, Slovakia in the field of information retrieval and neural networks.