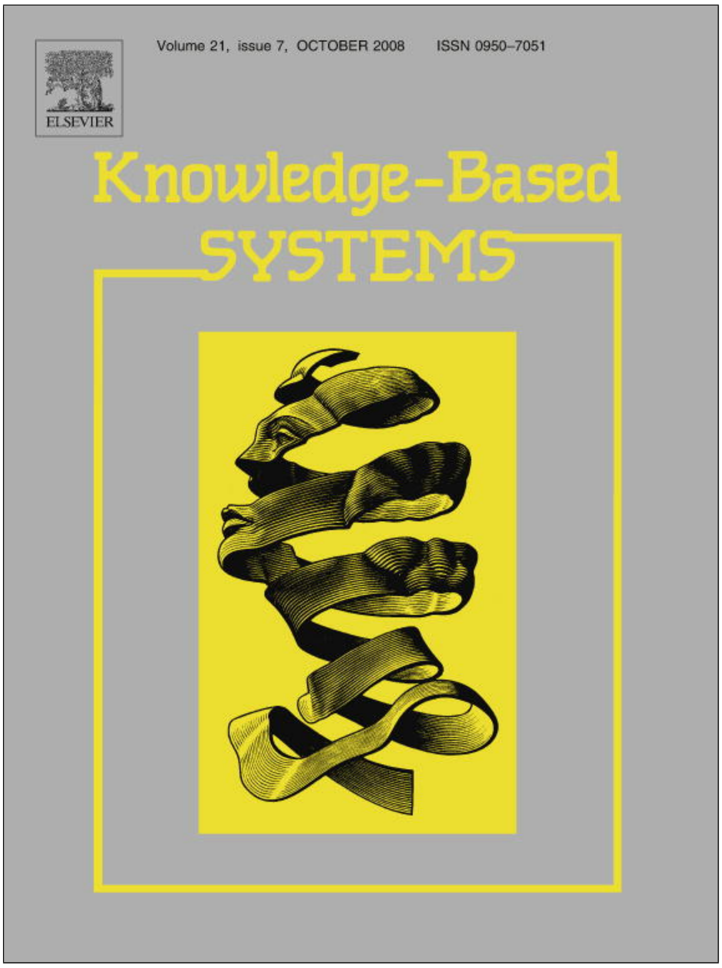
Volume 21, issue 7, OCTOBER 2008 ISSN 0950-7051
ELSEVIER
Knowledge-Based
SYSTEMS

Knowledge-Based Systems 21 (2008) 629–635

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Sentence recognition using artificial neural networks ☆

Maciej Majewski [a,*], Jacek M. Zurada [b]

[a] Faculty of Mechanical Engineering, Koszalin University of Technology, Raclawicka 15-17, 75-620 Koszalin, Poland
[b] Electrical and Computer Engineering Department, University of Louisville, 405 Lutz Hall, Louisville, KY 40292, USA

A R T I C L E   I N F O

Article history:
Received 17 June 2007
Accepted 24 March 2008
Available online 8 April 2008

PACS:
01.30.−y

Keywords:
Knowledge discovery
Sentence recognition
Artificial intelligence
Human–computer interaction
Natural language processing

A B S T R A C T

The paper describes an application of artificial neural networks (ANN) for natural language text reasoning. The task of knowledge discovery in text from a database, represented with a database file consisting of sentences with similar meanings but different lexico-grammatical patterns, was solved with ANNs which recognize the meaning of the text using training files with limited dictionary. The paper features recognition algorithms of text meaning from a selected source using 3-layer ANNs. Tests of the new method have also been described.

1. Introduction

For linguistic research, there is a need for consciously created and organized collections of data and information that can be used to perform knowledge discovery in text and to evaluate the performance and effectiveness of related tools. Knowledge discovery in text is a non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in unstructured text data. These patterns are unknown, hidden or implicit in semi-structured and unstructured collections of text. Below are some knowledge discovery tasks that many subject disciplines are interested in:

- Identification and retrieval of relevant documents from large collections of documents.
- Identification of relevant sections in large documents (passage retrieval).
- Co-reference resolution, i.e., the identification of expressions in texts that refer to the same entity, process or activity.
- Extraction of entities or relationships from text collections.
- Automated characterization of entities and processes in texts.
- Automated construction of ontologies for different domains (e.g., characterization of medical terms).
- Construction of controlled vocabularies from fixed sets of documents for particular domains.

The need to construct controlled vocabularies for subject domains has meant that terminological extraction from corpora has become an important process in tasks related to knowledge discovery in text.

The proposed system for knowledge discovery in text uses neural networks for natural language understanding as shown in Fig. 1. The motivation behind using the binary neural networks for knowledge discovery is that they offer an advantage of simple binarization of words and sentences, as well as very fast training and run-time response of this type of neural networks [14].

The system consists of a selected data source, 3-layer ANNs, network training sets, letter chain recognition algorithms, syntax analysis algorithms, as well as coding algorithms for words and sentences.

2. The state of the art

Knowledge discovery is a growing field. There are many knowledge discovery methodologies in use and under development. Some of these techniques are generic, while others are domain-specific.

☆ The source code of the implemented ANN is available by emailing the corresponding author.

* Corresponding author. Tel.: +48 94 3478352.
E-mail addresses: maciej.majewski@tu.koszalin.pl (M. Majewski), jacek.zurada@louisville.edu (J.M. Zurada).
URL: http://ci.louisville.edu/zurada (J.M. Zurada).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.03.053

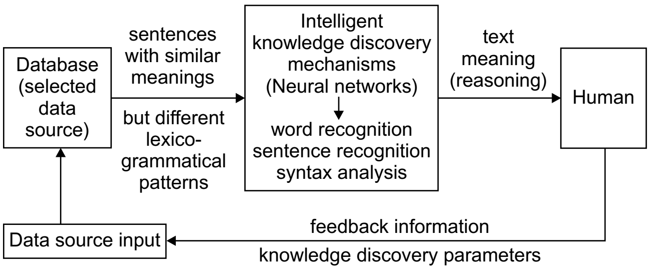

Fig. 1. Steps involved in proposed knowledge discovery in text.

Learning algorithms are an integral part of knowledge discovery. Learning techniques may be supervised or unsupervised. In general, supervised learning techniques enjoy a better success rate as defined in terms of usefulness of discovered knowledge. According to [1,3], learning algorithms are complex and generally considered the hardest part of any knowledge discovery technique. Machine discovery is one of the earliest fields that has contributed to knowledge discovery [5]. While machine discovery relies solely on an autonomous approach to information discovery, knowledge discovery typically combines automated approaches with human interaction to assure accurate, useful, and understandable results.

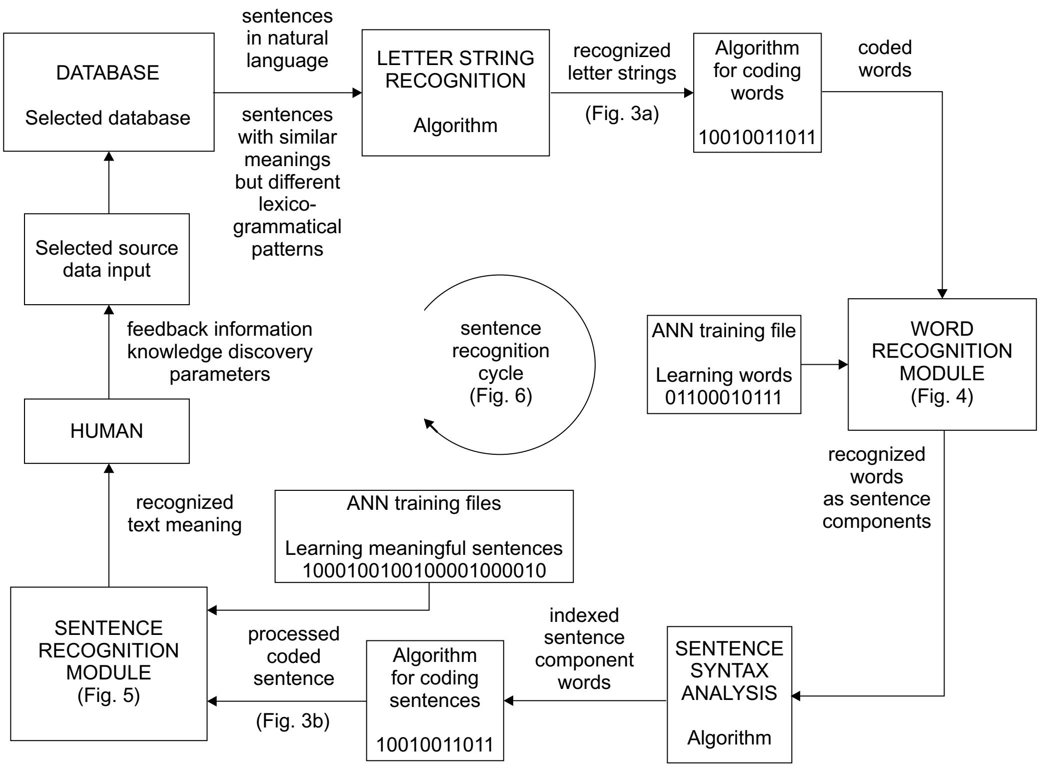

Fig. 2. Scheme of the proposed system for knowledge discovery in text.

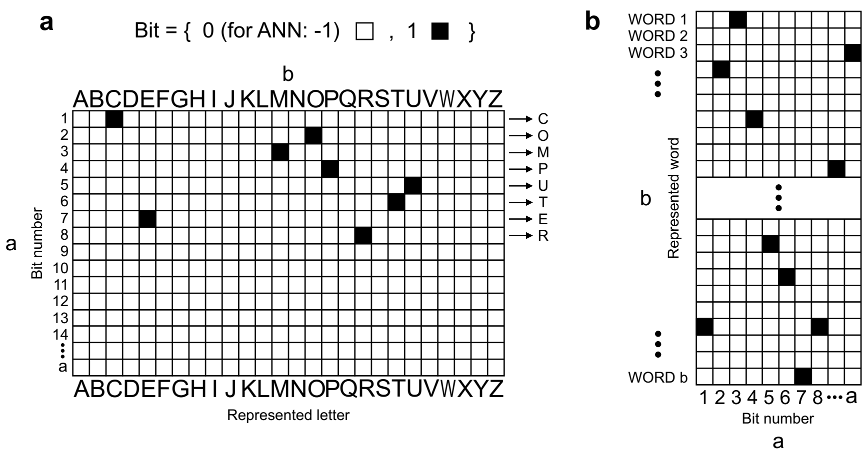

Fig. 3. Inputs of (a) the word recognition module, (b) the sentence recognition module.

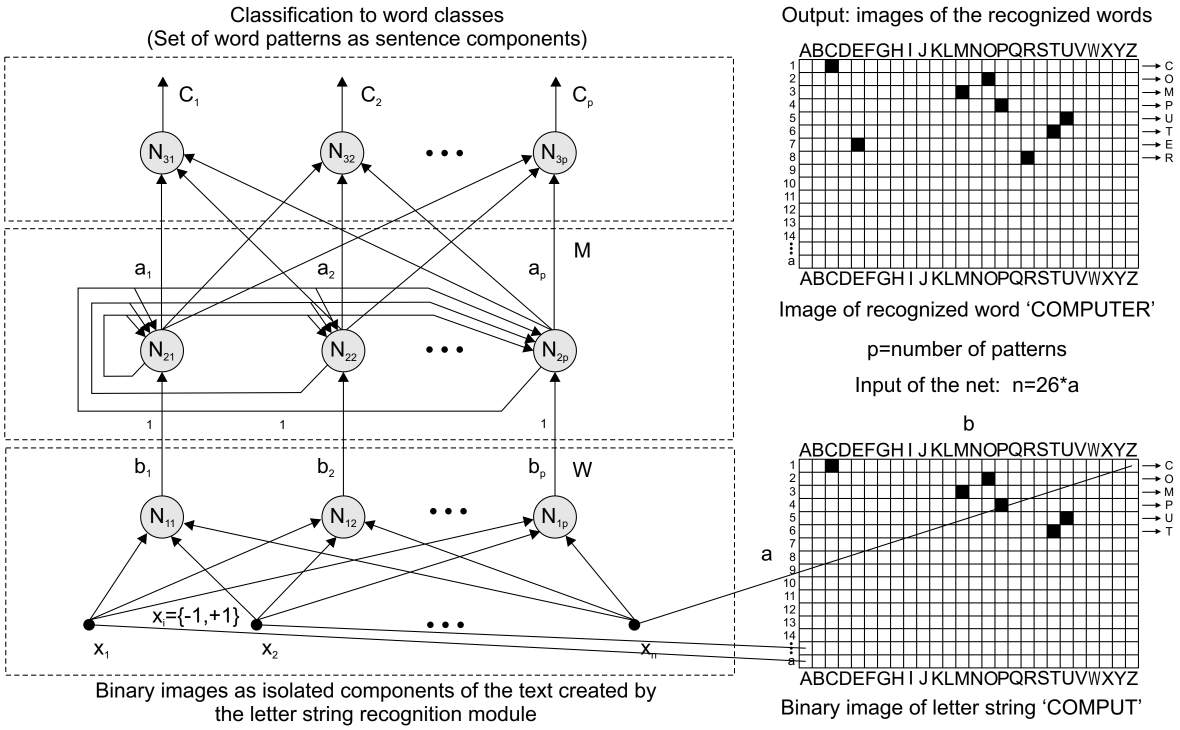

Fig. 4. Word recognition module.

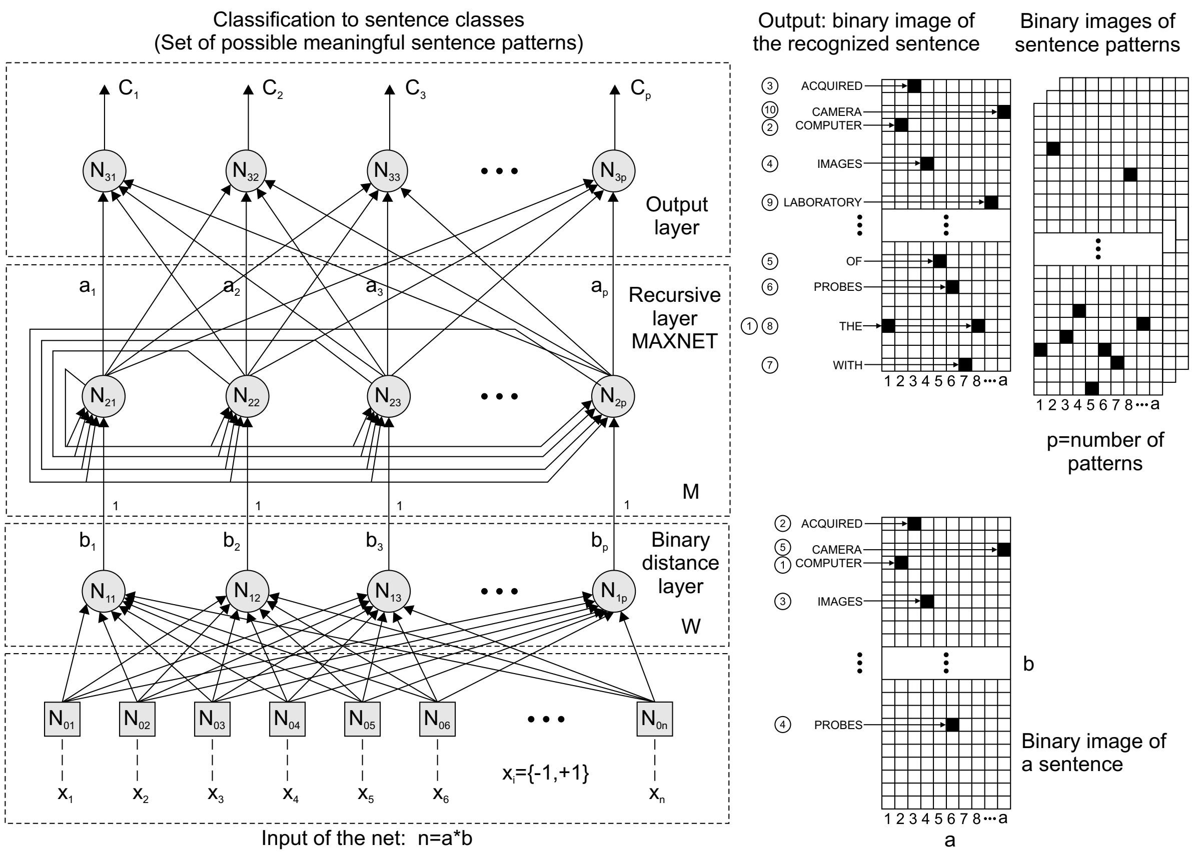

Fig. 5. Overall diagram of a 3-layer neural network for sentence recognition module.

There are many different approaches to knowledge discovery techniques [13]. There are quantitative approaches, such as the probabilistic and statistical approaches, other approaches utilize visualization techniques. There are classification approaches such as Bayesian classification, inductive logic, pattern discovery and decision tree analysis [3,5]. Classification is probably the oldest and most widely-used of all the knowledge discovery approaches [4,9,13]. This approach groups data according to similarities or classes. Pattern detection of important trends is the basis for the deviation and trend analysis approach. Other approaches include deviation and trend analysis, genetic algorithms, neural networks, and hybrid approaches that combine two or more techniques [8].

The probabilistic approach family of knowledge discovery utilizes graphical representation models to compare different knowl-

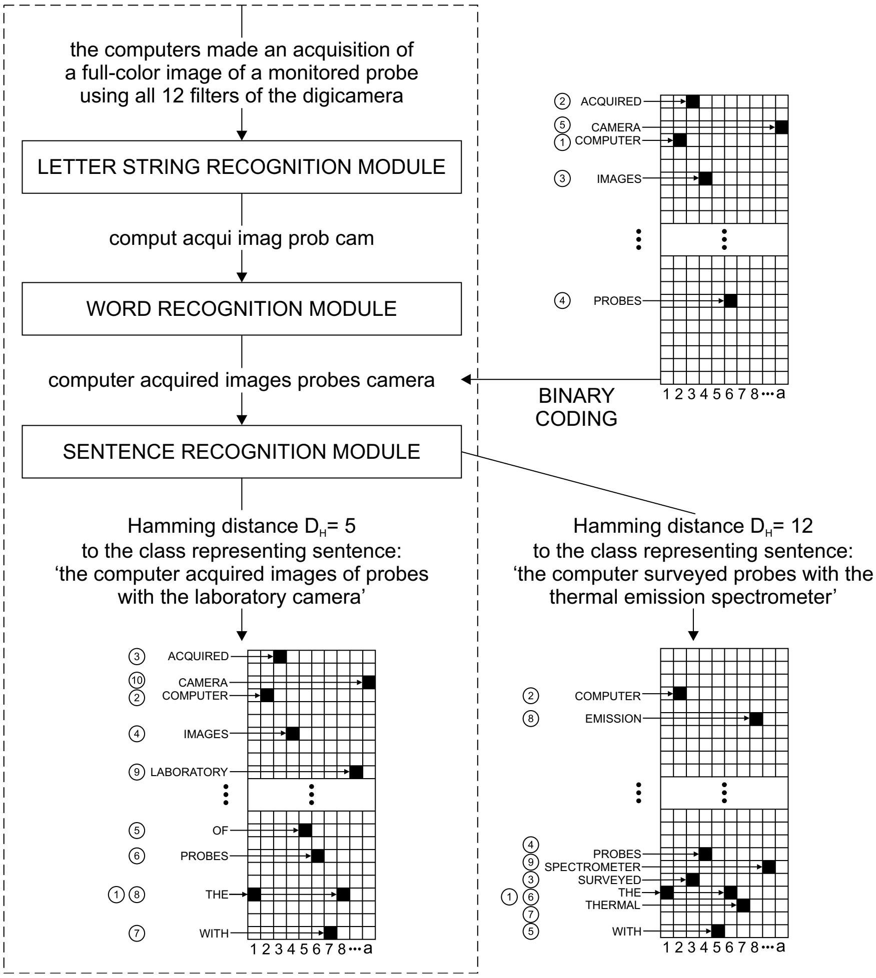

Fig. 6. Illustration of sentence recognition cycle.

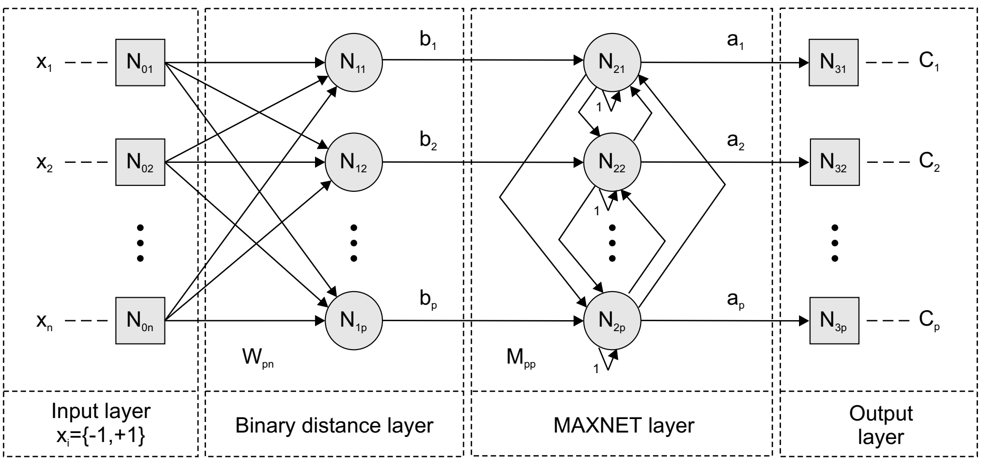

Fig. 7. Structure of the Hamming neural network as a classifier-expert module for word and sentence recognition.

edge representations [9]. These models are based on probabilities and data independencies. The statistical approach uses rule discovery and is based on data relationships. An inductive learning algorithm can automatically select useful join paths and attributes to construct rules from a database with many relations [4]. This type of induction is used to generalize patterns in data and to construct rules from the noted patterns.

ANNs may be used as tools for knowledge discovery. They are particularly useful for pattern recognition, and are sometimes grouped with the classification approaches. A hybrid approach to knowledge discovery combines more than one approach and is also called a multi-paradigmatic approach [2]. Although implementation may be more difficult, hybrid tools are able to combine their strengths of various approaches. Some of the commonly used methods combine visualization techniques, induction, neural networks, and rule-based systems to perform the desired knowledge discovery. Deductive databases and genetic algorithms have also been used in hybrid approaches.

The methods of sentence recognition proposed in the literature lead to interesting new approaches and techniques, but only very few experiments and evaluation of the proposed methods have been reported. In [6], an extended Kohonen feature map was described that was able to store sequences of input patterns. This network is able to learn to simulate a finite-state machine for the grammar, given examples of legal sentences from the regular grammar. Jiang in [7] proposed an algorithm which represents a framework for classifier combination in grammar-guided sentence recognition that is applicable to a variety of different tasks. In [12], the sentence recognition method using word cooccurrence probability was described, and compared with the method using Context-Free Grammar.

3. Description of the method

In the proposed knowledge discovery system shown in abbreviated form on Fig. 2, sentences are extracted from the database. Individual words treated here as isolated components of the text are processed by the letter string recognition and coding algorithms. The coded words are inputs of the neural network for recognizing words (Fig. 3a). The network uses a training file containing also words and is trained to recognize words as sentence components, with words represented by output neurons (Fig. 4). In the next stage, the coded words are transferred to the sentence syntax analysis module. The module analyses and indexes words properly before they are processed by the algorithm for coding sentences. The commands are coded as vectors and then become inputs of the sentence recognition module (Fig. 3b). The module uses a 3-layer Hamming neural network as shown in Fig. 5, either to recognize the sentence and find its meaning or else it fails to recognize it (Fig. 6). The neural network of this module uses a training file containing patterns of possible meaningful sentences.

Because of the binary input signals, the Hamming neural network is chosen for both the word recognition and sentence recognition module as shown in Fig. 7 which directly realizes the one-nearest-neighbour classification rule [15]. Each training data vector is assigned a single class and during the recognition phase only a single nearest vector to the input pattern x is found and its class C_i is returned. There are two main phases of the operation of the expert-network: training (initialization) and classification. Training of the binary neural network consists of copying reference patterns as the weights of the matrix W_{pn}, as follows (1):

$$w_i = x_i, \quad 1 \leqslant i \leqslant p \tag{1}$$

where p is the number of input patterns-vectors x, each of the same length n, w_i is the i-th row of the matrix W of dimensions $(p \times n)$. For given n the computation time is linear with the number of input patterns p.

The goal of the recursive layer N_2 is the selection of the winning neuron representing a word or command class. The characteristic feature of this group of neurons is a self connection of a neuron to itself with a weight $m_{ii} = 1$ for all $1 \leqslant i \leqslant p$, whereas all other weights are kept negative. Initialization of the layer N_2 consists of assigning negative values to the square matrix M_{pp} except the main diagonal. Originally Lippmann proposed initialization [10] (2):

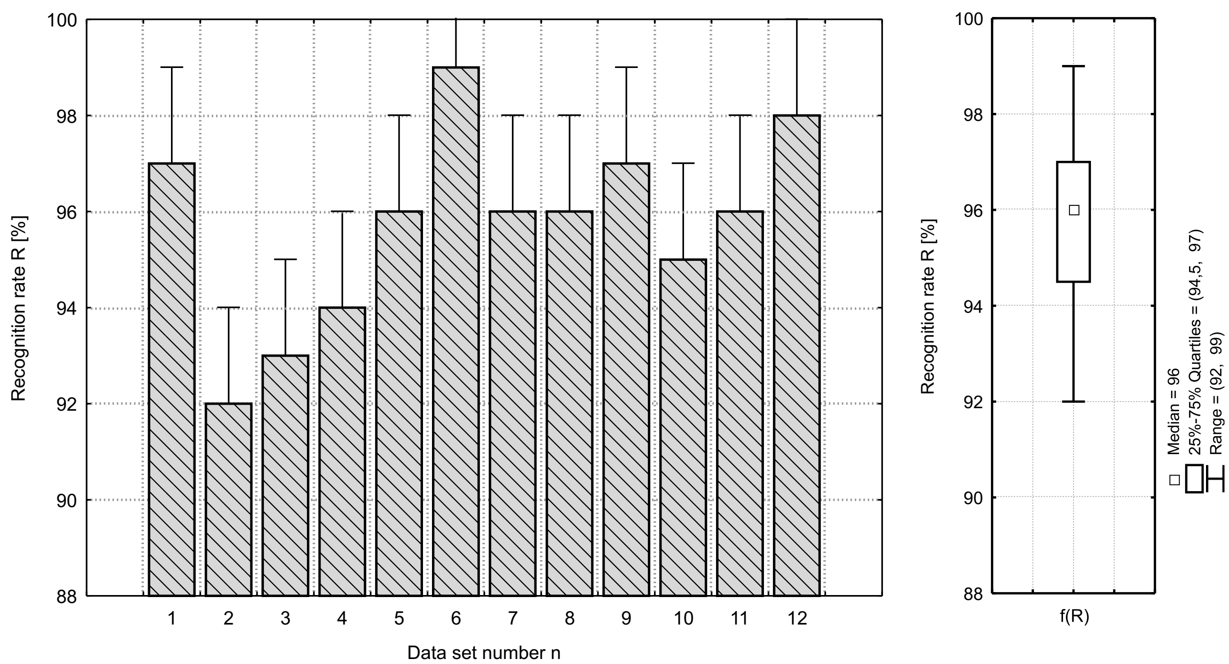

Fig. 8. Sentence meaning recognition rate as a set of words recognized earlier.

$$m_{kl} = -(p-1)^{-1} + \xi_{kl} \quad \text{for} \quad k \neq l, \quad 1 \quad \text{for} \quad k = l \qquad \text{where} \quad 1 \leqslant k, l \leqslant p, p > 1 \tag{2}$$

where ξ is a random value for which $|\xi| \ll (p-1)^{-1}$. However, it appears that the most efficient and still convergent solution is to set equal weights for all neurons in $N_{2.}$ which are then modified at each step during the classification phase as follows (3):

$$m_{kl} = \varepsilon_k(t) = -(p-t)^{-1} \quad \text{for} \quad k \neq l, \quad 1 \quad \text{for} \quad k = l \qquad \text{where} \quad 1 \leqslant k, l \leqslant p, p > 1 \tag{3}$$

where t is a classification time step. In this case the convergence is achieved in $p - 1 - r$ steps, where $r > 1$ stands for the number of nearest stored vectors in W.

In the classification phase, the group $N_{1.}$ is responsible for computation of the binary distance between the input pattern z and the training patterns already stored in the weights W. Usually this is the Hamming distance (4):

$$b_i(z, W) = 1 - n^{-1} D_H(z, w_i), \quad 1 \leqslant i \leqslant p \tag{4}$$

where $b_i \in [0, 1]$ is a value of an i-th neuron in the $N_{1.}$ layer, $D_H(z, w_i) \in \{0, 1, \ldots, n\}$ is a Hamming distance of the input pattern z and the i-th stored pattern w_i (i-th row of W).

In the classification stage, the layer $N_{2.}$ operates recursively to select the winning neuron. This process is governed by the following Eq. (5):

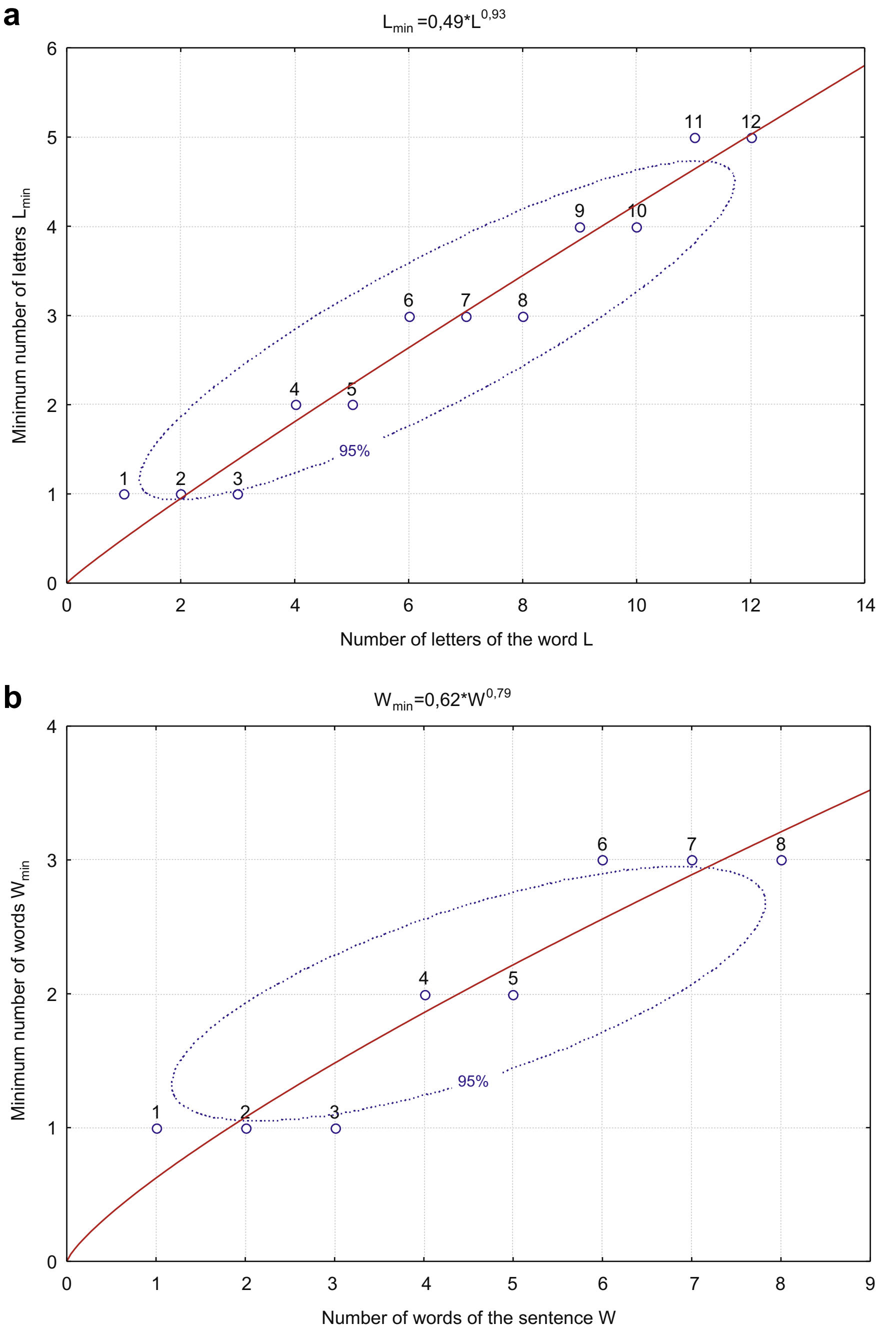

Fig. 9. Sensitivity of (a) word recognition: minimum number of letters of the word being recognized vs. number of word letters, (b) sentence meaning recognition: minimum number of words of the sentence being recognized vs. number of sentence component words.

$$a_i[t+1] = \varphi\left(\sum_{j=1}^{n} m_{ij}a_j[t]\right) = \varphi\left(a_i[t] + \sum_{j=1,i\neq j}^{n} m_{ij}a_j[t]\right) \quad (5)$$

where $a_i[t]$ is an output of the i-th neuron of the layer N_2 for the iteration t, φ is the threshold function defined as follows (6):

$$\varphi(x) = x \quad \text{for} \quad x > 0, \quad 0 \text{ otherwise} \quad (6)$$

Depending on the chosen scheme (2)–(3) of the weights m_{ij} in (5), we obtain different dynamics of the classification stage. The iterative process (5) proceeds up to a point where only one neuron has value different than 0 – this neuron is a winner which represents the word or command class.

4. Experimental results

The process of data insertion into the database of sentences probes for experiments was supported with the implemented speech interface. The speech recognition engine was a continuous density mixture Gaussian Hidden Markov Model system which uses vector quantization for speeding up the Euclidean distance calculation for probability estimation [11]. The system uses context dependent triphonic cross word acoustic models with speaker normalization based on vocal tract length normalization, channel adaptation using mean Cepstral subtraction and speaker adaptation using Maximum Likelihood Linear Regression.

The test dataset consisted of the database of 1500 sentences, files consisting of 522 letter strings, 87 training words and 510 meaningful training sentences. The first test measured the performance of the sentence meaning recognition with the sentence recognition module as a set of words recognized earlier (Fig. 8).

As shown in Fig. 9a, the ability of the implemented neural network to recognize a word depends on the number of letters of that word. For best performance, the neural network requires a minimum number of letters of each word being recognized as its input. As shown in Fig. 9b, the ability of the neural network to recognize the sentence depends on the sentence length (wordcount). Similarly, for best sentence recognition, the neural network requires a certain minimum wordcount of the given sentence.

5. Conclusions and perspectives

Application of binary neural networks allows for recognition of sentences in natural language with similar meanings but different lexico-grammatical patterns, which can be encountered in documents, texts, vocabularies and databases. The method presented in this paper can be easily extended.

In the literature there are very few reports about sentence recognition. The method proposed in this paper is a conceptually new approach to this problem. The experimental results of the proposed method of sentence recognition show its excellent and promising performance, and can be used for further development and experiments.

In the future, sentences in natural language will undoubtedly be the most important way of communication between humans and computers. Great progress is made in many fields of science, where communication between humans and computers is an important task. The proposed neural network is both effective and flexible which makes its applications possible. As an interface, it allows for more robustness to human's errors. The proposed solution also eliminates scarcities of the typical co-operation between humans and computers.

References

[1] M. Berry, G. Linoff, Mastering Data Mining, John Wiley & Sons, 2000.
[2] I. Cloete, J.M. Zurada, Knowledge-Based Neurocomputing, MIT Press, Cambridge, Massachusetts, 2000.
[3] M.H. Dunham, Data Mining Introductory and Advanced Topics, Prentice Hall, 2003.
[4] J. Han, M. Kamber, Data Mining: Concepts and Techniques, second ed., Morgan Kaufmann, 2006.
[5] D.J. Hand, H. Mannila, P. Smyth, Principles of Data Mining, MIT Press, 2000.
[6] A. Hoekstra, M. Drossaers, An extended Kohonen feature map for sentence recognition, in: Proceedings of ICANN 1993, Lecture Notes in Computer Science, Springer, 1993, pp. 404–407.
[7] X. Jiang, K. Yu, H. Bunke, Classifier combination for grammar-guided sentence recognition, in: Proceedings of MCS 2000, Lecture Notes in Computer Science, Springer, 2000, pp. 383–392.
[8] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, Wiley-IEEE Press, 2002.
[9] D.T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining, John Wiley & Sons, Inc., 2004.
[10] R. Lippman, An Introduction to Computing with Neural Nets, IEEE Transactions on Acoustic, Speech, and Signal Processing, IEEE Signal Processing Society, Piscataway 4 (3) (1987) 4–22.
[11] M. Majewski, W. Kacalak, Intelligent human–machine voice communication system, Engineering Mechanics International Journal for Theoretical and Applied Mechanics 12 (3) (2005) 193–200.
[12] I. Murase, S. Nakagawa, Sentence recognition method using word cooccurrence probability and its evaluation, in: Proceedings of ICSLP 1990, pp. 1217–1220.
[13] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Addison Wesley, 2005.
[14] J.M. Zurada, Introduction to Artificial Neural Systems, PWS Publishing Company, Boston, Massachusetts, 1992.
[15] J.M. Zurada, R.J. Marks, C.J. Robinson, Computational Intelligence: Imitating Life, IEEE Press, New York, 1994.