Neural Validation of Grammatical Correctness of Sentences

Dawid Połap
Institute of Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: Dawid.Polap@gmail.com

***Abstract*—In this paper, the idea of validating the correctness of sentences has been presented. In the proposed solution, traditional and modern theories of syntax grammar were used. In addition, preprocessing of sentences and neural approach to the problem has been shown. For different sentences the proposed method has been tested. Research results were presented and discussed in terms of the advantages and disadvantages of the presented method.**

I. Introduction

Artificial intelligence (AI) is becoming a major direction of today's science. Scientists of various fields biology, mathematics, physics and computer science try to improve and enrich the daily lives with new technologies that are the cornerstone to achieve true AI. Numerous application of methods known as AI allow us to choose more and more bolder science majors.

One of the main applications of these methods are image processing and pattern recognition. For example, in [1], [2], the authors proposed an alternative method of compressing digital images; in [3] the idea of using methods inspired by natural and physical phenomena to the problem of finding key points was explained. This problem has numerous applications in modern medicine, where key points on X-RAY images may represent a variety of diseases. The creation of decision support systems based on key points can help in diagnosis of difficulties to detect diseases and even prevented them by fast detection. In numerous scientific papers ([4], [5]) a possibility of using methods of computational intelligence to acquire such knowledge was shown, which in the next stage of research was used in decision support systems. For example, in [5] the use of knowledge of employees in the company to create meaningful groups of human resources in the company was discussed. Moreover, in the case of pattern recognition common problem is the number of samples (knowledge) required for the proper operation of the algorithm. Quite often, the number of these samples is just too small. In order to increase their numbers, many algorithms are developed for this purpose. In [6] was shown that the theory of fuzzy sets is a good option. Another solution is the use of computational intelligence [7]. Of course, the method of knowledge transfer for the various systems is also created, for instance for open and closed teacher-learner systems [8]. In [9] the possibility of using hardware design patterns in System-on-Chip designs was discussed. The frequent developing large applications require continuous supervision of changes and relationships. An important issue is to improve the changes in the existing code, for example, by finding and saving design patterns that have already been made [10].

Copyright © 2016 held by the authors.

Heuristic algorithms are widely used in other practical problems of modern computing as traffic optimization in digital databases where particle swarm algorithm was used [11]. In [12], the authors presents an application of neural networks to predict air pollution concentration for various chemical factors. The idea presents a model of virtual monitoring station points. Another increasing problem is the growing amount of data, and with it - the problem of queuing and database searches. Queuing is a problem that was born in the 50s of the last century with the development of the telecommunication networks. In the era of the Internet, the problem of queuing has grown - servers, databases, service providers, shops and websites are just a few examples where better and better solutions are needed today. In [13] the idea of the use of heuristic methods to positioning of queuing models was shown, and in [14], [15] the applicability of sorting algorithms for databases with a very large number of records. While [16] presents an algorithm to create mazes by artificial ant colony algorithm that may be used in two-dimensional games. Again in [17] was shown that use of computational intelligence as a manager in real-time cloud-based game management system where the algorithm modifies the game depending on the randomness and movement of the player can increase efficiency. As the final result, the method of creating games that scenarios are variables in real time was created so the game is different for each approach.

Natural language processing is another well developed area of computer science. In [18], the authors proposed the use of natural language processing in an accurate and efficient identification of problems related to opioid. Again in [19] presented a probabilistic framework that allows robots to perform your commands without knowing the environment. The authors tested and presented the results of their research on two mobile robots. Analysis of the authorship identification methods in the national language electronic discourse is presented in [20]. In the paper, an analysis of methods for English and discussed generalization of these methods for different languages were presented.

Moreover, there is more and more need for chat-bots, or question-answer systems or even systems that enable a conversation with a machine. In [21] was shown an innovative approach to natural language processing by the neural network composed of several parts - Sentence Layer, Layer Knowledge, Deep Layer Case and Network Dictionary. This approach allowed the authors to obtain first of all associative memory and a question-answering system. Natural language processing is quite complicated topic as evidenced by the numerous works of different approaches to the subject. [22] presented the idea of using image-text modeling, where the system was taught words using their representation. Moreover, the authors claim that their algorithm offers easy transfer to the sound. Many scientists from the whole world try to modify existing methods to get the best results eg.: convolution neural networks with little hyper-parameter tuning with numerous modifications [23] and in [24] shown the classic methods in this type of issue. An interesting algorithm for learning neural probabilistic language models was presented in [25]. The authors mentioned all the disadvantages of learning using gradient methods and large databases of words. They proposed algorithm based on noise-contrastive estimation. After testing the algorithm, they concluded that it is not only fast and effective but above all stable algorithm.

In this paper, two different approaches to automatic recognition of grammatically correct sentences are shown.

II. Theories of syntax and grammar

A. Traditional grammar

There are two different ways of notation in the grammar. The first one interprets the sentence as a combination of two elements - subject and predicate. This is one of the most common notation in English. To this day, it is called traditional grammar which comes from the Latin and Greek grammars. In this interpretation, we can identify some basic rules as

- The sentence is composed of subject and predicate.
- The predicate is regarded as a property or characteristic of the subject.
- The predicate must contain a verb.
- Verb in the predicate requires a specific object, predicative and adjuncts.
- The relation between the subject and the verb is called a nexus.

In [26], Otto Jespersen presented to the world the relation nexus in context of the analysis written text. Nexus has been described as a relation between the subject and his action in the sentence. Introduced relation was intended to create a system of symbols for parts of speech.

Let me take the following sentence to illustrate the nexus relation

$$\textit{His father is a king.} \qquad (1)$$

In proposed sentence *the father* is the subject, *is* refers to the action of the subjects and *the king* is the subject's predicate.

B. Modern grammar

Modern grammar is strongly associated with the work of the German mathematician Gottlob Frege. Frege was interested in the theory of predicate in the field of mathematical logic, which has become the cornerstone of new theories in linguistics - Frege argued that there is a distinction between sense and reference in relation to reference.

In this context, predicates of a sentence are treated as a mathematical functions. Predicates are used as a function that can assign multiple arguments to each other, so each sentence (in the linguistic sense) can be described mainly by predicates and their arguments. In [27], the author used this mathematical notation for linguistic sentences - applying this notation for sentence in (1), we get

$$is(\textit{His father}, \textit{a king}) \iff is = f(\textit{His father}, \textit{a king}). \qquad (2)$$

According to this model, the verb is the predicate and the noun phrase is an argument. In each sentence, all predicates create a matrix predicates, which comprises verbs, adjectives etc. Each sentence can be represented as a tree, where the words are vertices, as shown in Fig. 1, 2 and 3.

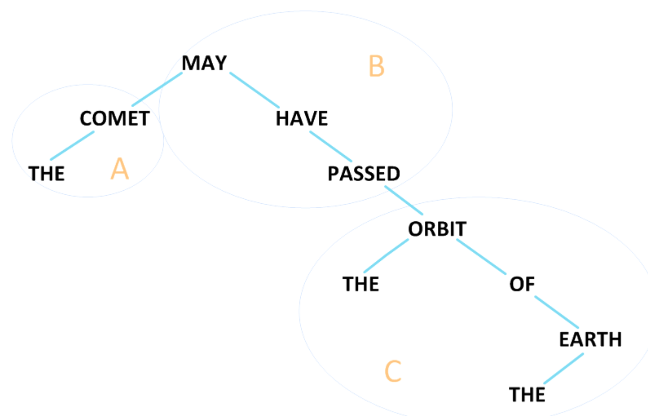

Fig. 1: Decomposition of the following sentence *The comet may have passed the orbit of the Earth*, where A is an argument of predicate, B means the matrix predicate and C represents the second argument of predicate (according to modern grammar).

III. Preprocessing sentences

To train the neural networks, an important step is proper preparation of samples. Neural network on the input accepts vector of numerical values, and the sentence in the sense of linguistics is not filed in numbers. To do this, we need to re-evaluate the words into numerical values. Moreover, the number of elements in samples is also important. The more the value, the longer the time of learning.

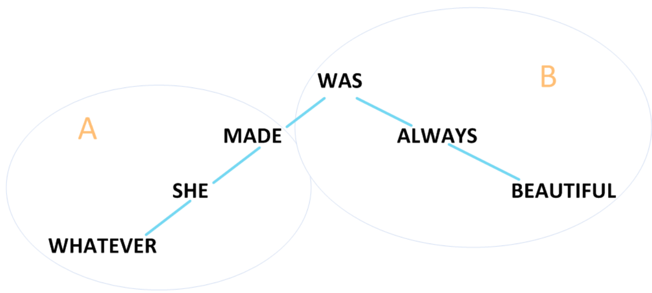

Fig. 2: Decomposition of the following sentence *Whatever she made was always beautiful*, where A is an argument of predicate and B means the matrix predicate (according to modern grammar).

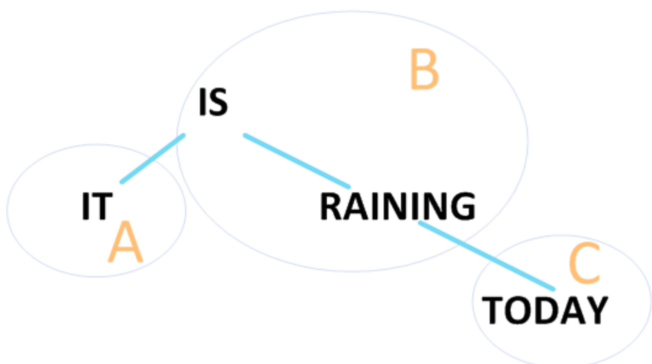

Fig. 3: Decomposition of the following sentence *It is raining today*, where A is an argument of predicate, B means the matrix predicate and C represents the second argument of predicate (according to modern grammar).

In order to reevaluate words in the numerical values used both grammatical notations - traditional (II-A) and modern (II-B).

A. Traditional notation

Algorithms type of part-of-speech tagging are related to finding specific parts of speech in a sentence. In the 60s of last century, the Brown Corpus (*Brown University Standard Corpus of Present-Day American English*) interested in this problem, created a huge database of words. It was not until 30 years later, in [28] used the hidden Markov models in this issue. The simplest method is to find probability of occurrence of a particular part of speech after the word. For example, if the current expression is *the*, the next most likely is a noun or an adjective. In subsequent years, in order to obtain higher accuracy, the number of analyzed words was increased - to three or four. Another known method of finding specific parts of speech is algorithm called VOLSUNGA presented in [29].

VOLSUNGA algorithm operates on the principle of finding the optimal path based on the matrix of probabilities. For each pair of words, the value p is calculated as the likelihood of the analyzed pair of parts of speech, what can be described by the following formula

$$p(p - pw) * p(pw, pa), \tag{3}$$

where w is the word, and pw means the part of speech and pa is a part of speech which may be the next word.

For all paths, algorithm selects the most optimal solution. In each iteration, a new word is added and the next optimal solution is searched. At the end of the algorithm, the most optimal path is returned, which represents the most likely sequence of parts of speech for the analyzed sentence.

In the case of traditional grammar, sentence must be distributed on the subject and predicate. Using the VOLSUNGA algorithm, we find a place of occurrence of the subject a and verb b. In next step, values a and b are re-calculated in accordance with

$$\begin{cases} val_1 = \sqrt{\frac{1}{a}\sum_{i=0}^{a} i^2} \\ val_2 = \sqrt{\frac{1}{b}\sum_{i=b}^{n} i^2} \end{cases}, \tag{4}$$

where n is the number of all words in a sentence, and val_1, val_2 mean the average amount of information contained in the subject and predicate. Using these values and the value c which takes the value 0 if sentence is incorrect or 1 in other case, a six-element vector can be created as

$$[val_1, val_2, a, b, n, c]\,. \tag{5}$$

Algorithm 1 VOLSUNGA algorithm

1: Start
2: Set probability matrix
3: **for each** pair of words **do**
4: Calculate the likelihood of analyzed pair using (3)
5: Select the most optimal solution
6: **end for**
7: Return the best solution
8: Stop

B. Modern grammar

For the grammar described in II-B approach, creating a sample is much easier than for a traditional approach - the problem is to find the matrix predicate. In the matrix predicate will be a verb and specify verb for each tense eg.: for the present perfect, it will be *have/has*. To find the matrix, we can find each verb by VOLSUNGA algorithm or find specific verb in the analyzed sentence and select the next verb.

After finding the matrix, vector representing the sentence can be created as

$$[val_3, val_4, val_5, val_6, k, c]\,, \tag{6}$$

where k means the number of arguments, and the remaining values are calculated by

$$\begin{cases} val_3 = \sum_{j=0}^{k} \sqrt{\frac{1}{c} \sum_{i=0}^{c_j} i^2} \\ val_4 = \sqrt{\frac{1}{b} \sum_{i=0}^{c_m} i^2} \\ val_5 = \frac{1}{k} \sum_{j=0}^{k} c_j \\ val_6 = \frac{c_m}{k} \end{cases}, \quad (7)$$

where c_j is the number of words in j-th argument, c_m is the number of words in the matrix. Values val_3, val_4 represent average amount of information in arguments/matrix and val_5, val_6 are arithmetic amount of words respectively in the argument and the matrix.

C. Example

Consider the following sentence

Whatever she made was always beautiful. (8)

In the case of traditional notation, we obtain the following values

$$val_3 = \sqrt{\frac{1}{2} * 5} \approx 1,6$$

$$val_4 = \sqrt{\frac{1}{3} * 86} \approx 5,4,$$

which comprise the following vector

$$[1,6;5,4;2;3;6;1].$$

In the case of modern notation, we obtain

$$val_3 = \sqrt{\frac{1}{6} * 5} \approx 0,9$$

$$val_4 = \sqrt{\frac{1}{3} * 30} \approx 3,2$$

$$val_5 = 2 * \frac{1}{6} = 0,33$$

$$val_6 = 4 * \frac{1}{6} \approx 0,17$$

and finally

$$[0,9;3,2;0,33;0,17;6;1]$$

IV. Neural validation

In order to validate the correctness of sentences, a neural network was used ([30]–[32]) with back propagation algorithm. Neural networks are mathematical models inspired by the operation of the neural network in the human brain. Model of the entire network can be divided into three types of layers: an input, hidden and output. Each layer consists many neurons where each of them is connected to all the neurons in adjacent layers. The connection between two neurons is marked by weight. Weights are selected in a random way at the beginning of the algorithm.

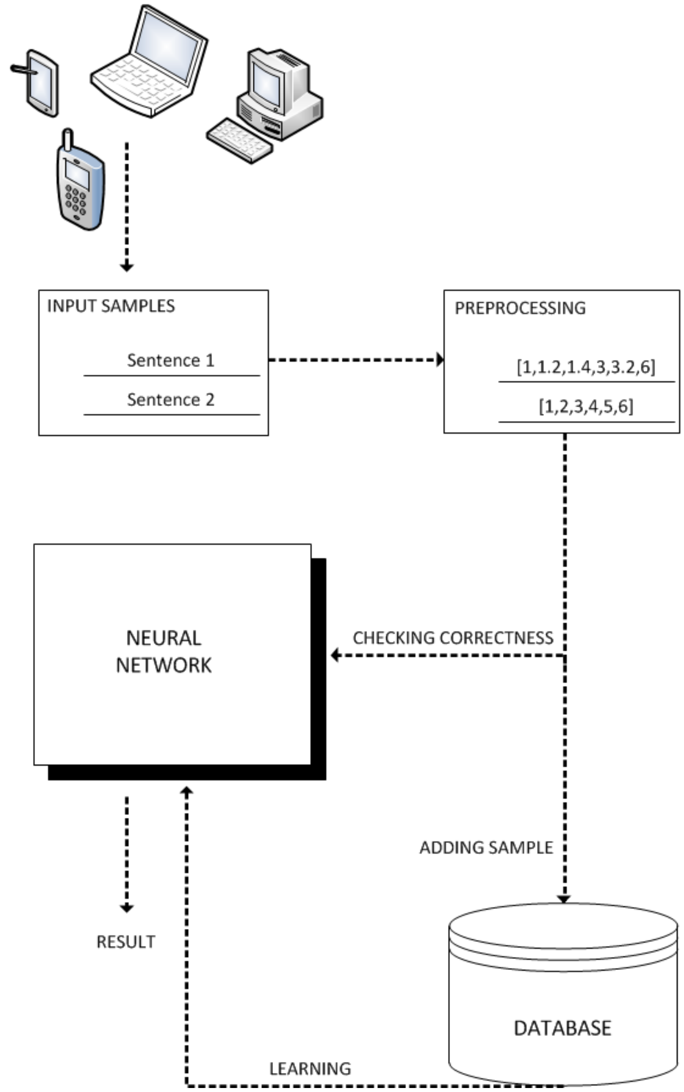

Fig. 4: The model of the proposed system to validate grammatical sentences.

Each neuron is composed of the value (signal y) of each neuron from the previous layer and weights w imposed on the connection between each pair of neurons. In the neuron, each signal is multiplied by the weight and all the values are summed. The value of the sum of products is understood as a neuron argument. The output value of the neuron is the value of the activation function. The sigmoid function is one of the most popular which value is in the range $\langle 0,1 \rangle$ and the formula is

$$f(x) = \frac{1}{1+e^{-\beta x}}, \quad (9)$$

where β is a parameter and x is the neuron argument calculated by

$$x = \sum_{i=0}^{n} w_i y_i, \quad (10)$$

where n is the number of neurons in the previous layer. In the input layer, learning vectors (samples) are entering signals.

In the hidden layer, neurons recalculate values obtained from previous layers and the output layer returns the result of the network.

For the purposes of validation of sentences, proposed network is constructed of 6 neurons in the input layer, 3 neurons in each of the 3 hidden layers and 1 neuron in the output layer.

A. The Backpropagation Algorithm

The Backpropagation Algorithm ([33], [34]) is used to modify weights in the neural network to get the most accurate learning. The algorithm assumes the calculation of error of each neuron starting at the output layer until the first hidden layer. Using the calculated error, the weight of the connection between the neurons is respectively modified by

$$w = w + \Delta w, \tag{11}$$

where Δw is an error calculated by

$$\begin{cases} \Delta w = f(x)(1 - f(x)(p - f(x)) & \text{for the output layer} \\ \Delta w = f(x)(1 - f(x)) \sum\limits_{signals} w_i \delta_i & \text{for the hidden layer} \end{cases} \tag{12}$$

where p is expected output.

Algorithm 2 The Backpropagation Algorithm

```
Start
Generate random weight
while global_error>0.1 do
    for each layer in network do
        for each neuron in layer do
            Calculate the neuron activation in accordance with
            (9) and (10)
        end for
    end for
    for each neuron in output layer do
        Calculate error using (12)
    end for
    for each hidden layer do
        for each neuron in hidden layer do
            Calculate error using (12)
            Modify the weight using (11)
        end for
    end for
end while
Stop
```

V. Experiments

For the purpose of checking the correctness of the proposed validation, a database of 200 sentences was created. Every sentence was processed in two vectors one for each grammar. The neural network was trained for $\beta = 0,6$. In the next step, each vector was processed by a trained neural network to check the quality of the validation. The obtained results are shown in Tab. I. The graph showing the error reduction over the number of epochs is in Fig. 5 [18].

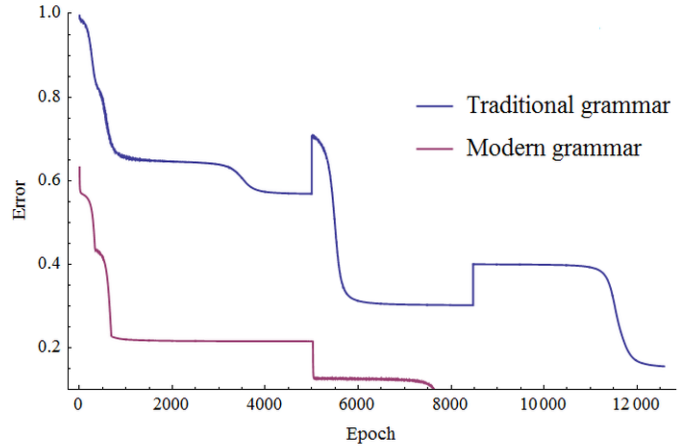

Fig. 5: Sample neural network learning process error.

TABLE I: The effectiveness of the network for the prepared samples.

Grammar	Effectiveness
Traditional	67%
Modern	89%

VI. Conclusions

The proposed system allows us to check whether a sentence is grammatically correct. This is an important aspect for the development of a communication system between man and machine. The proposed solution includes two grammatical notations. The results show that the use of modern theories of language based on the logic of predicates can be crucial in further developing the field of natural language processing.

In future research work, we plan to further develop the topic of natural language (including analysis of words and sentences).

Acknowledgment

Author acknowledge contribution to this project of Operational Programme: “Knowledge, Education, Development” financed by the European Social Fund under grant application POWR.03.03.00-00-P001/15.

References

[1] F. Hussain and J. Jeong, “Efficient deep neural network for digital image compression employing rectified linear neurons,” *Journal of Sensors*, vol. 2016, 2015.

[2] M. Wozniak, C. Napoli, E. Tramontana, and G. Capizzi, “A multiscale image compressor with rbfnn and discrete wavelet decomposition,” in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[3] M. Woźniak and Z. Marszałek, “An idea to apply firefly algorithm in 2d image key-points search,” in *Proceedings of International Conference on Information and Software Technologies*. Springer, 2014, pp. 312–323.

[4] C. Napoli and E. Tramontana, “An object-oriented neural network toolbox based on design patterns,” in *Proceedings of International Conference on Information and Software Technologies (ICIST)*. Springer, 2015, pp. 388–399.

[5] C. Napoli, G. Pappalardo, E. Tramontana, R. K. Nowicki, J. T. Starczewski, and M. Woźniak, "Toward work groups classification based on probabilistic neural network approach," in *Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer, 2015, pp. 79–89.

[6] B. A. Nowak, R. K. Nowicki, M. Woźniak, and C. Napoli, "Multi-class nearest neighbour classifier for incomplete data handling," in *Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer, 2015, pp. 469–480.

[7] M. Wozniak, D. Polap, G. Borowik, and C. Napoli, "A first attempt to cloud-based user verification in distributed system," in *Proceedings of Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*. IEEE, 2015, pp. 226–231.

[8] R. Damaševičius, "Towards empirical modelling of knowledge transfer in teaching/learning process," in *Proceedings of Information and Software Technologies*. Springer, 2014, pp. 359–372.

[9] R. Damaševičius, G. Majauskas, and V. Štuikys, "Application of design patterns for hardware design," in *Proceedings of the 40th annual Design Automation Conference*. ACM, 2003, pp. 48–53.

[10] S. Cicciarella, C. Napoli, and E. Tramontana, "Searching design patterns fast by using tree traversals," *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 321–326, 2015.

[11] M. Wozniak, "On positioning traffic in nosql database systems by the use of particle swarm algorithm," in *Proceedings of XV Workshop DAGLI OGGETTI AGLI AGENTI–WOA*, 2014, pp. 25–26.

[12] G. Capizzi, G. L. Sciuto, P. Monforte, and C. Napoli, "Cascade feed forward neural network-based model for air pollutants evaluation of single monitoring stations in urban areas," *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 327–332, 2015.

[13] M. Woźniak, W. M. Kempa, M. Gabryel, R. K. Nowicki, and Z. Shao, "On applying evolutionary computation methods to optimization of vacation cycle costs in finite-buffer queue," in *Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer, 2014, pp. 480–491.

[14] M. Woźniak, Z. Marszałek, M. Gabryel, and R. K. Nowicki, "Modified merge sort algorithm for large scale data sets," in *Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC)*. Springer, 2013, pp. 612–622.

[15] Z. Marszalek, M. Wozniak, G. Borowik, R. Wazirali, C. Napoli, G. Pappalardo, and E. Tramontana, "Benchmark tests on improved merge for big data processing," in *Proceedings of Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*. IEEE, 2015, pp. 96–101.

[16] D. Połap, "Designing mazes for 2d games by artificial ant colony algorithm," in *Proceedings of Symposium for Young Scientists in Technology, Engineering and Mathematics*, 2016, pp. 63–70.

[17] D. Połap, M. Wozniak, C. Napoli, and E. Tramontana, "Real-time cloud-based game management system via cuckoo search algorithm," *International Journal of Electronics and Telecommunications*, vol. 61, no. 4, pp. 333–338, 2015.

[18] D. S. Carrell, D. Cronkite, R. E. Palmer, K. Saunders, D. E. Gross, E. T. Masters, T. R. Hylan, and M. Von Korff, "Using natural language processing to identify problem usage of prescription opioids," *International journal of medical informatics*, vol. 84, no. 12, pp. 1057–1064, 2015.

[19] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz, "Inferring maps and behaviors from natural language instructions," in *Proceedings of Experimental Robotics*. Springer, 2016, pp. 373–388.

[20] A. Venčkauskas, R. Damaševičius, R. Marcinkevičius, and A. Karpavičius, "Problems of authorship identification of the national language electronic discourse," in *Proceedings of Information and Software Technologies*. Springer, 2015, pp. 415–432.

[21] T. Sagara and M. Hagiwara, "Natural language neural network and its application to question-answering system," *Neurocomputing*, vol. 142, pp. 201–208, 2014.

[22] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proceedings of International Conference on Machine Learning (ICML)*, 2014, pp. 595–603.

[23] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[24] Y. Chen, "Convolutional neural network for sentence classification," 2015, university of Waterloo.

[25] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426*, 2012.

[26] O. Jespersen, *Analytic syntax*. University of Chicago Press, 1937.

[27] D. J. Allerton, *Essentials of grammatical theory: A consensus view of syntax and morphology*. Routledge, 1979.

[28] J. Kupiec, "Robust part-of-speech tagging using a hidden markov model," *Computer Speech & Language*, vol. 6, no. 3, pp. 225–242, 1992.

[29] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," *Comput. Linguist.*, vol. 14, no. 1, pp. 31–39, Jan. 1988. [Online]. Available: http://dl.acm.org/citation.cfm?id=49084.49087

[30] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[31] J. E. Dayhoff, *Neural network architectures: an introduction*. Van Nostrand Reinhold Co., 1990.

[32] D. Psaltis, A. Sideris, and A. Yamamura, "A multilayered neural network controller," *IEEE control systems magazine*, vol. 8, no. 2, pp. 17–21, 1988.

[33] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1989, pp. 593–605.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.