UDC 004.896

AUTOMATED METHODS OF COHERENCE EVALUATION OF UKRAINIAN TEXTS USING MACHINE LEARNING TECHNIQUES

A.A. Kramov, S.D. Pogorilyy

Taras Shevchenko National University of Kyiv

The main methods of coherence evaluation of texts with the usage of different machine learning techniques have been analyzed. The principles of methods with the usage of recurrent and convolutional neural networks have been described in details. The advantages of a semantic similarity graph method have been considered. Other approaches to perform the vector representation of sentences for the estimation of semantic similarity between the elements of a text have been suggested to use. The experimental examination of methods has been performed on the set of Ukrainian scientific articles. The training of recurrent and convolutional networks with the usage of early stopping has been performed. The accuracy of the solving of document discrimination and insertion tasks has been calculated. The comparative analysis of the results obtained has been performed.

Keywords: coherence of a text, recurrent neural network, convolutional neural network, semantic similarity graph, semantic representation of sentences, document discrimination task, insertion task.

Проаналізовано основні методи оцінки когерентності текстів з використанням різних технологій машинного навчання. Детально описано принципи роботи методів з використанням рекурентної та згорткової нейронних мереж, розглянуто їх переваги та недоліки. Обгрунтовано доцільність використання методу графу семантичної схожості порівняно з іншими методами. Запропоновано використання інших підходів векторного представлення речень для розрахунку міри семантичної схожості елементів тексту. Проведено експериментальну перевірку проаналізованих методів на множині україномовних наукових статей, здійснено навчання моделей семантичного представлення слів та речень. Виконано навчання рекурентної та згорткової нейронних мереж з використанням методу раннього зупину. Обраховано точність вирішення задач розрізнення документів та вставки для проаналізованих методів, здійснено порівняльний аналіз отриманих результатів.

Ключові слова: когерентність тексту, рекурентна нейронна мережа, згорткова нейронна мережа, граф семантичної схожості, семантичне представлення речень, задача розрізнення документів, задача вставки.

Introduction

According to the permanent increase of the amount of heterogeneous data in an information space, it is necessary to consider the automatic analysis of data with unfixed structure: texts, audio records, images, video, etc. Taking into account the need to search for appropriate data among this space, it is advisable to pay attention to the processing of text information that is performed by search engines in order to find resources that are relevant to user's queries (texts, audio, and video files). Moreover, the automated analysis of a text is used in different linguistic tasks, data extraction algorithms, recognition tasks, text generation, etc. Both mentioned tasks and other issues connected with the processing of text information should fall into a category of *natural language processing* (NLP) problems. The heterogeneity of text structure makes it impossible to use a universal common approach for the solving of NLP tasks. Due to the growth of the power of computational resources, it becomes advisable to use different machine learning models pre-trained on a corresponding set of text information. Such an approach is used in order to solve NLP tasks that fall into a category of AI-complete issues, i.e. tasks that cannot be solved without human computation. *The coherence evaluation of a text* belongs to this kind of task.

The coherence of a text implies its thematic integrity and communication purpose to convey its key idea to a reader [1]. Moreover, the coherence of a text provides the structural integrity of a text (its cohesion). Text that is more coherent is easier to understand due to its ordered structure, the availability of presupposition and implication elements that are connected to general world knowledge. Fig. 1 demonstrates the simplified example of coherent and incoherent text fragments. In contrast to the coherent fragment that performs information representation consequently, the incoherent fragment contains a sentence (2) which content does not correspond to the whole topic of the snippet. Despite the availability of the connection between the first and the second sentences of the incoherent fragment (e.g. the coreferent connection "Ann"-"she" [2]), the content of the sentence (2) deviates from the common topic (daily routine) that complicates the perception of the fragment.

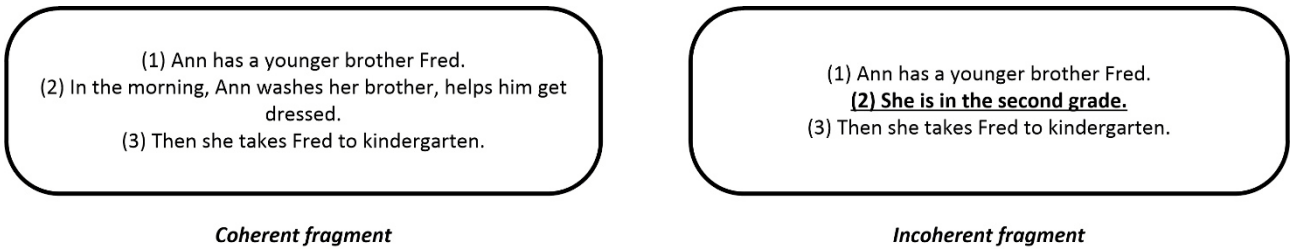

Fig. 1 – The examples of the coherent and incoherent parts of a text

The automated estimation of text coherence can be utilized in different NLP areas:

- search engines and SEO analysis;
- copywriting;
- creation of educational material;
- detection of the symptoms of mental illness [3].

The relevance of the coherence estimation task may be explained by the availability of state-of-the-art works [3; 4; 5; 6] devoted to the creation of methods of coherence evaluation of text information in order to solve different issues. It should be mentioned that most corresponding papers are related to the analysis of English texts. Despite the availability of Ukrainian research papers devoted to the solving of NLP tasks, the investigation of the coherence evaluation of Ukrainian-language texts is still at the initial stage. Thus, the usage of different coherence estimation methods for a Ukrainian corpus requires further analysis and experimental verification.

The purpose of the paper consists in the performing of the comparative analysis of state-of-the-art coherence evaluation methods for English texts and the implementation of the experimental verification of the effectiveness of different methods (with its modification) based on machine learning models for Ukrainian corpora.

The automated methods of the coherence evaluation of a text

Different methods of the coherence evaluation of texts utilize machine learning models: neural networks, support vector machines, decision trees, etc. It should be mentioned that the usage of some machine learning models is complicated due to the unfixed structure of input data: texts may incorporate any amount of words and sentences. Moreover, it is advisable to perform the formalization of text elements taking into account is semantic, syntactic, or spatial properties. Let us consider the coherence estimation methods based on the analysis of the semantic component of a text in a detail.

Distributed sentence representation method based on the usage of recurrent neural network. *A recurrent neural network* [7] is used in different NLP tasks. Such a choice can be explained by the ability of this network architecture to process data sequences with variable length: signals are passed to the input of the neurons of a recurrent layer in a recursive manner. The recurrent neural network of different architecture (for instance, *long short-term memory, LSTM*) uses internal memory to process the sequences of signals of any length. Moreover, such signal flow through recurrent layers reproduce the perception of a text by a reader: neurons process both a current signal and inputs retrieved at the previous stage. This signal processing corresponds to the analysis of a text by a human because brain neurons perceive information according to the previously read fragment [8]. Fig. 2 shows the example of signal flow through a recurrent layer.

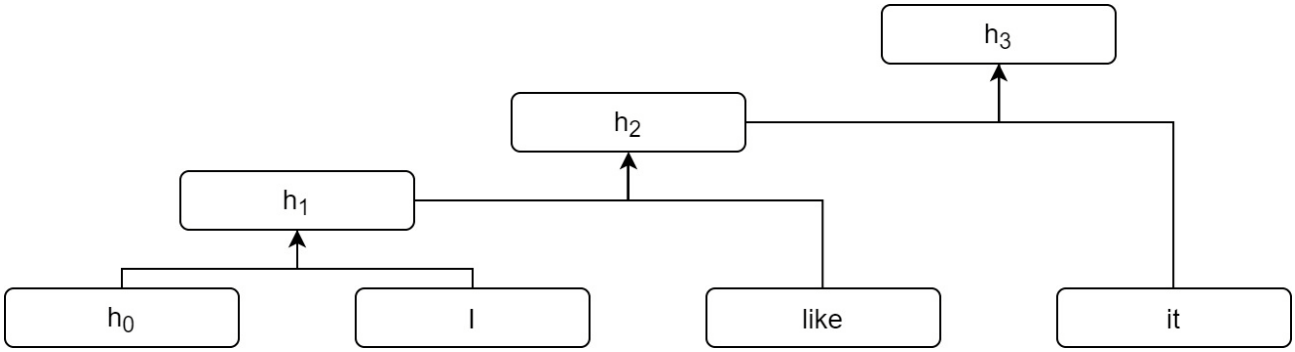

Fig. 2 – The example of signal (word) flow through a recurrent layer

Firstly, the preprocessing of an input text is performed: tokenization operation (representation of a text as a list of sentences and their words) is applied. Then the formalized representation of words is performed, namely, the representation of their semantic component. Such formalization is made with the usage of a pre-trained embedding model. For instance, Word2Vec [9] or GloVe [10] models may be utilized. Thus, each sentence is represented in the following way:

$$s = \left\{ \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_s} \right\}, \quad (1)$$

where n_s is a count of sentence words; $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{n_s}$ – vector representation of corresponding words. Retrieved vectors are consequently passed to the input of the recurrent layer. The output value of the layer $\mathbf{h}_t$ at the discrete time step is calculated by the following equation:

$$\mathbf{h}_t = f(V_{\text{Recurrent}} \cdot \mathbf{h}_{t-1} + W_{\text{Recurrent}} \cdot \mathbf{w}^t + \mathbf{b}_{\text{Recurrent}}), \quad (2)$$

where $V_{\text{Recurrent}}$ and $W_{\text{Recurrent}}$ are free parameters; $\mathbf{b}_{\text{Recurrent}}$ is a bias vector; f denotes a non-linear activation function. After the performing of the vector representation of each sentence, the groups of sentences (cliques) with a fixed length L are formed according to their order and ordinary step (for instance, $\langle s_1, s_2, s_3 \rangle, \langle s_2, s_3, s_4 \rangle$).

Then the vector representation of each clique is implemented by applying a concatenation operation to the vectors of an input clique. A retrieved vector $\mathbf{h}_c$ is passed to the input of a binary classifier that consists of several dense layers. The last layer incorporates a single neuron with a softmax activation function; thus, the output of this neuron represents the coherence value of the input clique. The coherence of a whole document D is calculated as a product of the coherence estimations of all cliques:

$$T_D = \prod_{c \in D} y_c \quad (3)$$

Coherence evaluation method based on the usage of a convolutional neural network. Convolutional neural networks [7] are utilized in the different tasks related to the processing of input images or video. Such a choice can be explained by the following factors:

- the possibility to process input data that are represented in a matrix form;
- the availability of a multichannel architecture that allows performing the parallel processing of several data threads (for instance, RGB channels [11]).

However, the possibility to process input data with an unfixed structure and the availability of several separate channels allow utilizing convolutional layers for NLP tasks [12]. In contrast to the recurrent neural network, a convolutional network does not provide taking into account word order while processing input sequence; however, the availability of separate channels allows the extension of the structure of a network in order to analyze other properties of a text (e.g. syntactical or spatial features).

The preprocessing of a text is performed in the same manner as for the previous method: each sentence is represented as a set of vectors (see equation (1)) after the applying of a pre-trained embedding model. Moreover, an input document is transformed into a set of cliques; the coherence value of the document is calculated as the product of the coherence values of corresponding groups (see equation (3)). Let us consider the process of the coherence evaluation of each clique.

The formalized representation of each clique's sentence is performed in a matrix form: the columns of a matrix are formed from the vector representations of sentence words. Each formed matrix is passed to the input of a separate channel; thus, a count of the channels of an input convolutional layer defines the size of a clique. Fig. 3 shows the example of the applying of a convolutional operation to sentences matrices in order to form corresponding feature vectors (separate vectors for each channel).

Firstly, the extraction of features from input data is performed by the convolutional layer with the usage of a set of filters (matrices $F \in \mathrm{R}^{d \times m}$ where m denotes the width of a sliding "window", d is the dimension of the vector representation of sentence words). Each filter slides across the columns of a matrix S with an ordinary step; the result of the execution of each step can be represented as a vector $c \in \mathrm{R}^{|S|-m+1}$. The components of the vector are calculated in the following way:

$$c_i = \left(S * F\right)_i = \sum_{k,j} \left(S_{[i-m+1:i]} \otimes F \right)_{kj}, \quad (4)$$

where $\otimes$ denotes the element-wise multiplication of vectors. The output of the convolutional layer is represented as a set of feature maps C. The next component of the network is a pooling layer that performs the aggregation of the retrieved set C: a maximum value is chosen from each map. Thus, feature vectors are formed for each input sentence. Then the concatenation of the feature vectors of all channels is performed; the retrieved common vector $\mathbf{h}_c$ is passed to the sequence of dense layers that performs the estimation of the coherence of an input clique. The signal flow through these layers is made in the same manner as for the previously considered method.

The graph-based method with the usage of a semantic similarity graph. In contrast to the previously considered methods, a method with the usage of a semantic similarity graph provides the visualization of the process of the formation of a coherence estimation. Such an approach allows tracking the algorithm of the calculation of an output result with a further modification of a text in order to increase the coherence value.

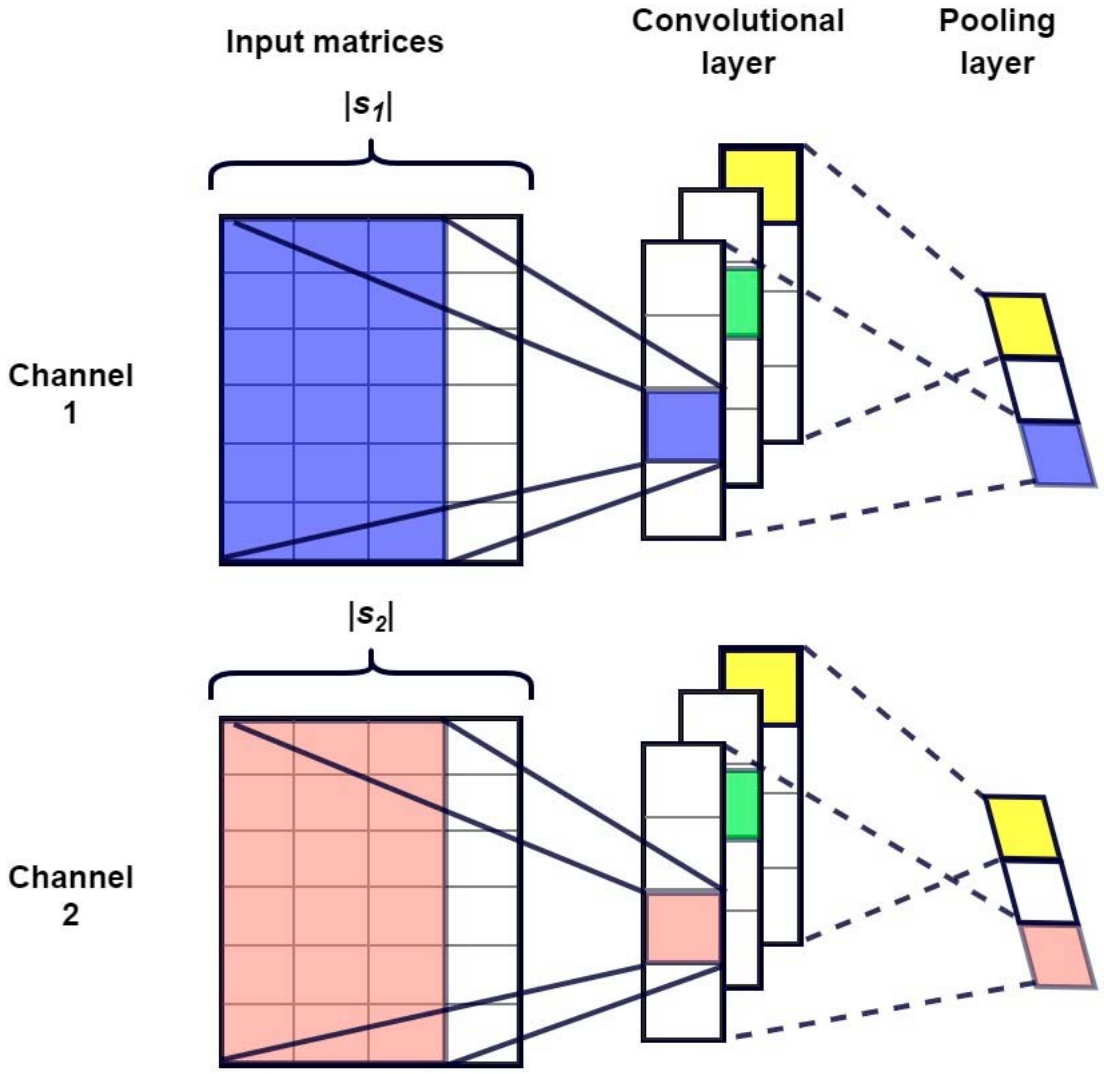

Fig. 3 – The applying of a convolutional operation in order to form feature vectors (multichannel processing)

The method is based on the building of a directed graph $G(V,E)$, where V is a set of vertices that correspond to text sentences; E is a set of edges. The weight of edges interprets the semantic similarity measure of adjacent vertices (sentences). The formalization of sentences is performed by the vector representation of their words; a corresponding vector according to a pre-trained embedding model represents each word. Thus, a sentence s should be considered as a set of word vectors:

$$s = \left\{ \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M \right\}, \tag{5}$$

where M is a count of sentence words. The vector representation of the whole sentence s is performed as an average value of corresponding vectors:

$$\mathbf{s} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{w}_k \tag{6}$$

In order to set edges and calculate their weights, three different approaches can be utilized: PAV, SSV, and MSV. In the case of the usage of the *PAV* approach, the verification of the possibility to set an edge between a current vertex and the previous one (i.e. vertex that represents the previous sentence) is made. If a similarity measure between

these vertices is equal to zero than an attempt to set an edge between the current vertex and the next previous vertex is executed. Thus, the outdegree value of each vertex does not exceed one. The similarity measure of sentences s_i and s_j is calculated in the following way:

$$\mathrm{sim}(s_i, s_j) = \alpha uot(s_i, s_j) + (1-\alpha)\cos(\mathbf{s}_i, \mathbf{s}_j), \quad (7)$$

where uot denotes the ratio of a number of common entities over a number of all entities of the sentences s_i and s_j; $\cos(\mathbf{s}_i, \mathbf{s}_j)$ is a cosine distance between corresponding sentence vectors; α is a regulative parameter, $\alpha \in [0,1]$.

In the case of the *SSV* approach, the search for the most similar vertex according to a defined similarity measure is performed for each vertex. An edge is set just between a current vertex and the most similar one. Thus, the outdegree value of vertices does not exceed one too. The similarity measure of the sentences s_i and s_j is calculated in the following way:

$$\mathrm{sim}(s_i, s_j) = \frac{\cos(\mathbf{s}_i, \mathbf{s}_j)}{|i-j|} \quad (8)$$

As for the *MSV* approach, edges are set between all vertices if the weight value of an edge exceeds a preset threshold value θ. The weight of edges is calculated by the equation (8).

The coherence estimation of a text D for all approaches is calculated as the average value of the edges' weights of the built graph $G(V, E)$. Fig. 4 demonstrates the example of graphs for considered approaches.

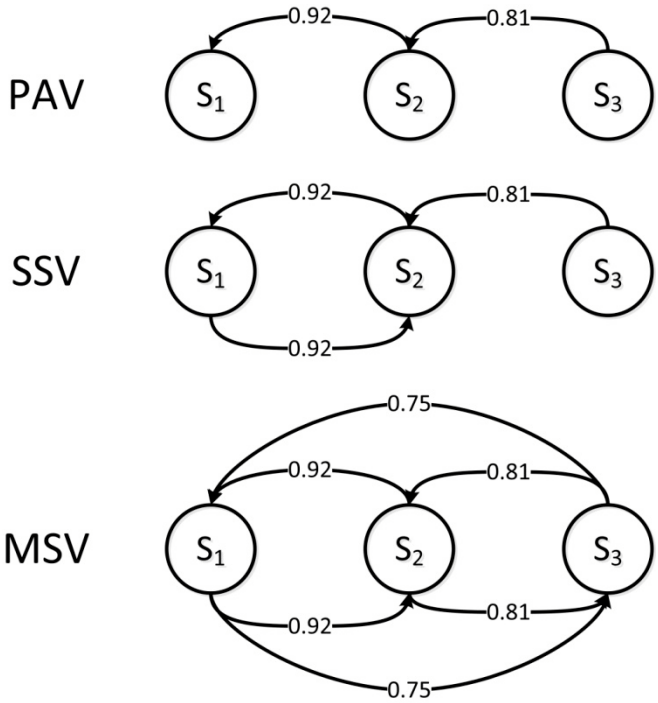

Fig. 4 – The examples of a semantic similarity graph $G(V, E)$ for different approaches

The experimental verification of the methods of the coherence evaluation for Ukrainian texts

It was decided to verify the effectiveness of the considered methods on a set of Ukrainian texts. The process of the experimental verification should be divided into the following consequent stages:

- collecting Ukrainian texts;
- preparation of input data for the training of neural networks;
- training of models;
- calculation of metrics that represent the effectiveness of the methods of the coherence evaluation of texts.

Collecting Ukrainian texts. Scientific articles from the websites of journals were used in order to generate training and validation sets of Ukrainian texts. Such a choice can be explained by the availability of the certain structure of scientific articles, the lack of phraseology, and the concise logical representation of the content of papers. In order to perform automatic extraction of data from the web pages of scientific journals, the Trinity method was utilized [13]. The automated analysis of 266 websites of Ukrainian scientific journals of different areas ("Applied sciences",

"Humanities", etc.) was performed. The abstracts of papers were extracted in order to form the training dataset. The test dataset was formed from the full versions of articles. The automatic detection of Ukrainian-language papers was implemented with the usage of a software module *langdetect* [14]. The full versions of articles are available just in PDF format; in order to extract the text of a paper, a client-server application Science Parse [15] was utilized. Taking into account the lack of the universal structure of the papers of different journals and the error of the Science Parse application, the extracted texts were additionally processed by a separate software module for the removing of incorrect symbol sequences (fragments without semantic meaning): tables, figure captions, lists, stop words, etc.

Preparation of input data for the training of neural networks. The result of the previous collecting of Ukrainian texts is the sets of abstracts and the full versions of Ukrainian-language articles. The training of a semantic embedding model should be considered as the preprocessing stage for all considered methods. The Word2Vec model was chosen as the embedding model. In contrast to the full versions of articles, the abstracts were extracted from HTML pages; therefore, they do not contain incorrect symbol sequences. Moreover, the abstracts contain main terms according to the topic of a paper. Thus, it was decided to form a training set for the Word2Vec model from the set of extracted abstracts. A lemmatization operation (transformation of a word to its normal form) was applied for the mentioned set. The formed training corpus was also utilized for the learning process of another embedding model, namely, Doc2Vec [16] (DBOW, DM, DBOW+DM) that performs the vector representation of text fragments. Such an approach allows both avoiding the formation of the vector representation of a sentence by calculating the average value of corresponding word vectors (see equation (6)) and taking into account word order within a sentence.

In order to train recurrent and convolutional neural networks, two sets (training and test) were formed from the set of full articles. A validation set was generated from the training set for the further prevention of the overfitting of the networks. The formation of input cliques was made with the following preprocessing stages:

- tokenization – the representation of a text as a list of sentences and their words;
- lemmatization;
- generation of coherent/incoherent cliques by the permutation of sentences within each clique.

The lemmatization and tokenization of texts were implemented using the utilities of the *lang.org.ua* web resource [17].

Training of models. The software implementation of the considered methods and corresponding neural networks was performed by the creation of an application written in Python 3.6. The training of neural networks Word2Vec, DBOW, DM, DM+DBOW was implemented using the embedded classes of the module *genism* [18] with the following parameters:

- vector dimension – 300;
- epoch count – 50;
- "window" size – 10;
- threshold value of frequency vocabulary – 1.

The software library Keras [19] was utilized for the designing and training of recurrent and convolutional neural networks. The training was done in 20 epochs using an early stopping approach in order to avoid overfitting and to save the best checkpoint of networks.

Calculation of metrics that represent the effectiveness of the methods of the coherence evaluation of texts. In order to estimate the metrics of the effectiveness of the methods of the coherence evaluation of texts, the accuracy of the solving of two tasks was calculated: *document discrimination task* and *insertion task*. The accuracy of the solving of these tasks is calculated in the following way:

$$acc = \frac{CR}{TR}, \tag{9}$$

where CR is a number of recognized texts; TR is a general number of texts. The difference between methods consists in the choice of the criterion whether this text was recognized or not. In the case of the solving of the document discrimination task, the permutation of sentences within a text is performed; the document is considered as recognized if the coherence value of its original version is higher than the coherence of the changed one. According to the insertion task, the following operation is made: some sentence is picked up randomly with the further removing from the text. Then the insertion of the selected sentence is performed into all possible positions except the native one. The document is considered as recognized if the coherence value of the original version is higher than the coherence of each new version of a text.

Table 1 shows the calculated accuracies for the document discrimination task and the insertion task retrieved by the different methods of the coherence evaluation of Ukrainian-language texts with different approaches (embedding models, regulative parameters).

The highest value of the accuracy of the solving of the document discrimination task was obtained with the method based on the convolutional neural network [20]. However, the accuracy of the solving of the insertion task for the methods based on neural networks is lower in comparison with the graph-based method. Such a result can be explained by the approach of the formation of the training dataset for neural networks that someway represents the document discrimination task. The formation of cliques according to the insertion task requires the increase of the training corpus and free parameters. The highest value of the accuracy of the solving of this task was obtained by the method based on the semantic similarity graph with the MSV approach, $\theta = 0$ [21]. A zero value of the threshold

parameter and the usage of the MSV approach may indicate the necessity to take into account the connection between all sentences in spite of their semantic similarity measure. Moreover, the optimal value of the regulative parameter $\alpha = 0.8$ for the PAV approach may underline the expediency of the analysis of common sentence elements: coreferent pairs and same words.

Table 1

The accuracy of the solving of the document discrimination task and the insertion task using different methods of the coherence evaluation for the Ukrainian-language corpus

Method	Approach	Embedding model	Document discrimination task, %	Insertion task, %
Recurrent neural network	–	Word2Vec	80.0	9.0
Convolutional neural network	–	Word2Vec	**99.0**	13.0
Semantic similarity graph	PAV, $\alpha = 0.8$	Word2Vec	58.0	41.0
	SSV	Word2Vec	55.0	22.0
	MSV, $\theta = 0$	Word2Vec	70.0	51.0
	PAV, $\alpha = 0.8$	DBOW	62.0	40.0
	SSV	DBOW	55.0	22.0
	MSV, $\theta = 0$	DBOW	78.0	62.0
	PAV, $\alpha = 0.8$	DM	62.0	37.0
	SSV	DM	50.0	20.0
	MSV, $\theta = 0$	DM	80.0	**66.0**
	PAV, $\alpha = 0.8$	DBOW+DM	69.0	35.0
	SSV	DBOW+DM	63.0	22.0
	MSV, $\theta = 0$	DBOW+DM	70.0	32.0

Conclusions

The methods of the coherence evaluation of texts based on the usage of different neural networks and graph theory have been analyzed. The usage of convolutional and recurrent layers allows the processing of input data with unfixed length: words or sentences. The expediency of the applying of the recurrent layer consists in taking into account the word order of a sentence that reproduces the perception of a text by a reader in a word-by-word manner. The advantage of the usage of the convolutional layer is the ability to perform the parallel processing of input data by means of a multichannel structure. In contrast to the mentioned methods, the method based on a semantic similarity graph utilizes neural networks just for the vector representation of sentences, therefore, it is possible to track the process of the formation of an output result and to analyze possible variants how to increase the coherence estimation value of a text.

The experimental verification of the effectiveness of the considered methods with different approaches for a set of Ukrainian-language scientific articles has been performed. The results obtained for the document discrimination task and the insertion task may indicate that the semantic similarity graph can be used in order to estimate the coherence of Ukrainian texts. The values of regulative parameters may point to the necessity to take into account the semantic and lexical components between all sentences. The accuracy of the methods may be improved by means of the increase of the training set and free parameters; moreover, it is advisable to consider other syntactical and spatial features of a text in order to retrieve the higher values of corresponding coherence estimation metrics.

References

1. Lednik, O. (2010). Cohesion and coherence as a category of cohesive text. *Scientific journal of M.P. Dragomanov National Pedagogical University. Series 10: Problems of grammar and lexicology of the Ukrainian language*. [Online]. (6). pp. 119–123. Available from: http://enpuir.npu.edu.ua/handle/123456789/15909. [Accessed: 23 January 2020].
2. Pogorilyy, S. & Kramov, A. (2019). Coreference Resolution Method Using a Convolutional Neural Network. In: *Proceeding of the 2019 IEEE International Conference on Advanced Trends in Information Theory*. 2019, pp. 397-401. Available from: [Accessed: 20 February 2020].
3. Bedi, G., Carrillo, F., Cecchi, G., Slezak, D., Sigman, M., Mota, N., Ribeiro, S., Javitt, D., Copelli, M. & Corcoran, C. (2015). Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia*. 1 (1). Available from: [Accessed: 23 January 2020].

4. Cui, B., Zhang, Y. & Zhang, Z. (2017). Text Coherence Analysis Based on Deep Neural Network. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, Singapore, pp. 2027–2030. Available from: [Accessed: 23 January 2020].
5. Li, J. & Hovy, E. (2014). A model of coherence based on distributed sentence representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 2039–2048. Available from: [Accessed: 23 January 2020].
6. Giray, G. & Ünalır, M. (2019). Assessment of text coherence using an ontology□based relatedness measurement method. *Expert Systems*. Available from: [Accessed: 23 January 2020].
7. Haykin, S. (2016). *Neural Networks: A Comprehensive Foundation Second Edition*. 2nd Ed. Kyiv.
8. Pogorilyy, S., Kramov, A. & Yatsenko, F. (2019). A method for analyzing the coherence of Ukrainian-language texts using a recurrent neural network. *Mathematical machines and systems*. 4. pp. 9–16. Available from: [Accessed: 23 January 2020].
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. 2013, pp. 3111–3119. Available from: [Accessed: 23 January 2020].
10. Pennington, J., Socher, R. & Manning, D. (2014). GloVe: Global Vectors for Word Representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543. Available from: [Accessed: 23 January 2020].
11. Cui, Z., Henrickson, K., Ke, R., Pu, Z. & Wang, Y. (2019). Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. In: *IEEE Transactions on Intelligent Transportation Systems*. 2019, pp. 1–12. Available from: [Accessed: 23 January 2020].
12. Kim, Y. (2014). Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1746–1751. Available from: [Accessed: 23 January 2020].
13. Pogorilyy, S. & Kramov, A. (2018). Automated extraction of structured information from a variety of web pages. In: *Proceedings of the 11th International Conference of Programming UkrPROG 2018*. 2018, pp. 149–158. Available from: [Accessed: 23 January 2020].
14. Nakatani, S. (2010). [Online]. 2010. Language Detection Library for Java. Available from: https://code.google.com/archive/p/language-detection. [Accessed: 23 January 2020].
15. AI2 (2020). *allenai/science-parse*. [Online]. 2020. GitHub. Available from: https://github.com/allenai/science-parse. [Accessed: 23 January 2020].
16. Le, Q. & Mikolov, T. (2014). Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. 2014, pp. 1188–1196. Available from: [Accessed: 23 January 2020].
17. Anon (2020). *Homepage: lang-uk*. [Online]. 2020. Lang.org.ua. Available from: http://lang.org.ua. [Accessed: 23 January 2020].
18. Řehůřek, R. & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. 2010, pp. 45-50. Available from: [Accessed: 23 January 2020].
19. Anon (2020). *Home - Keras Documentation*. [Online]. 2020. Keras.io. Available from: https://keras.io. [Accessed: 23 January 2020].
20. Pogorilyy, S., Kramov, A. & Biletskyi, P. (2019). Method for coherece evaluation of Ukrainian texts using convolutional neural network. *Collection of scientific works of the Military Institute of Kyiv National Taras Shevchenko University*. (64). pp. 5-13. Available from: [Accessed: 23 January 2020].
21. Pogorilyy, S. & Kramov, A. (2018). Method of the coherence evaluation of Ukrainian text. *Data Recording, Storage & Processing*. 20 (4). pp. 64–75. Available from: [Accessed: 23 January 2020].

About authors:

Kramov Artem Andriiovych,
postgraduate student (Computer Engineering); Ukraine, Taras Shevchenko National University of Kyiv.
A number of publications in Ukrainian journals: 8.
A number of publications in foreign journals: 1.
h-index: 1.
ORCID: https://orcid.org/0000-0003-3631-1268.

Pogorilyy Sergiy Demianovych,
Doctor of Sciences, Professor, Head of Computer Engineering Department; Ukraine, Taras Shevchenko National University of Kyiv.
A number of publications in Ukrainian journals: 280.
A number of publications in foreign journals: 50.
h-index: Google Scholar – 7, Scopus – 2.
ORCID: https://orcid.org/0000-0002-6497-5056 .

Authors' place of work:
Taras Shevchenko National University of Kyiv
Faculty of Radio Physics, Electronics and Computer Systems
01601, Kyiv, Volodymyrska str., 64/13,
E-mail: sdp77@i.ua, artemkramovphd@knu.ua

Контактна інформація для роботи з редактором:
E-mail: artemkramov@gmail.com
Телефон: +380501493132

Назва статті:
Автоматизовані методи оцінки когерентності україномовних текстів з використанням методології машинного навчання
Automated methods of coherence evaluation of Ukrainian texts using machine learning techniques

Автори:
Крамов Артем Андрійович, Погорілий Сергій Дем'янович
Kramov Artem Andriiovych, Pogorilyy Sergiy Demianovych