UDC 681.3

**O.O. MARCHENKO, O.S. RADYVONENKO, T.S. IGNATOVA,
P.V. TITARCHUK, D.V. ZHELEZNIAKOV**

IMPROVING TEXT GENERATION THROUGH INTRODUCING COHERENCE METRICS

Abstract. Text-based interaction using mobile devices is now ubiquitous, its main outlets being social networks, messengers, email conversations, virtual assistants, accessibility applications, etc. Its status implies the need to facilitate text input by the user and to devise ways to provide verbal feedback. In this paper, we discuss a method of unique text generation for mobile devices and its evaluation methodology as a solution for both stated challenges. We consider the opportunities given by the use of context (location, weather, scheduled events, etc.), the limitations in terms of computational resources and data usage, and the inherent subjectivity of creative task assessment given the number variety of possibly acceptable outputs. The comparison with other text generation approaches shows that the use of coherence metrics helps to achieve higher quality in terms of human perception. The Spearman correlation between the values of the proposed coherence metric and the human assessment of text readability is 0.86, which indicates the high quality of the metrics and the effectiveness of the method as a whole.

Keywords: natural language processing, automatic natural language text generation, coherency, coherence metrics.

INTRODUCTION

In this paper, we consider the task of automatic generation of short coherent text comprising several sentences based on a query or a given set of input keywords. A short chunk of text is given as an input, and we expect a two or three sentence long unique text as a result. The output text should be related to the input query without resorting to directly repeating or paraphrasing it, as can be seen in examples below.

On query “Attended, rock concert” the system should generate “We attended the concert together. I went to a rock and roll concert for the first time in my life”. On query “Cat, window” we expect to read something like “The cat practically lives in front of the window. She loves watching birds in summer.”

Promising results have been achieved in generating sentences of short and even medium length. There are various effective approaches for measuring text coherency, allowing to assess and control the level of coherency. However, generating longer coherent texts is still an open problem.

Based on the most effective methods of measuring the coherence of texts, we try to develop principles, approaches, and methods for the formation of coherent texts from individual sentences. But while structural methods as explicit ones can be researched and analyzed from within, identifying features and developing selection heuristics, dissecting neural network systems in the similar vein is almost impossible due to their “black box” nature and interpretability issues.

The main purpose of this paper is to research and develop effective methods for constructing coherent texts on request without using lexical-semantic knowledge bases specially prepared for this task. Given the above, we reformulate the task:

Problem. There is a set of correct sentences $S = S_1, S_2, \ldots, S_N$.

It is necessary to form a coherent text consisting of 2 or 3 sentences relevant to a query Q by choosing from the S a set of the most appropriate instances (with possible subsequent modification of these sentences at the syntax level, where necessary).

Text Generation. The most promising approach to implementing such systems is currently the use of neural networks, for example, convolutional neural networks, recurrent networks, recursive networks, and their various combinations for learning word sequences from the sentences in the train data set. After training, a neural network has to be able to generate sentences that form a unique coherent text.

Neural networks have been widely used to solve a number of problems of computational linguistics where it is necessary to find correspondences between pieces of text written by people, according to some sets of features, such as authorship attribution [1] and paraphrase identification [2]. Neural networks are also able to adequately generate very short messages for interactive systems [3, 4]. GPT-2 [5] model allows generating texts of significant length, but the size of the model makes its use very limited. At the moment, there are still difficulties with generating a single coherent text. The challenges can be generally demonstrated using the example of two well-known approaches: "skip thoughts" [6] and Sequence to Sequence Recurrent Neural Network [7]. When the "skip thoughts" model is used, a chain of semantic connections between word meanings is broken beyond the sequence of length n, where word $n+1$ is dissonant with the previous sequence (both between sentences and within one sentence, if it is longer than n). Recurrent neural networks tend to loop, and often generate sentences that are almost completely identical to the query and to each other by semantic meaning and lexical content, which cannot be considered a cohesive and coherent text.

Problems such as the curse of dimensionality make it hard to use exclusively neural network approaches to solve the problem of generating complete coherent texts. Hybrid systems that employ the principles of structural algorithmic natural language processing along with neural networks make it possible to use the advantages of neural network technologies, compensating their weak points by optimizing data, heuristics, special metrics, etc.

Structural methods for generating texts were shown to be useful in a number of works [8–10]. Despite the well-known advantages, they have two serious interconnected drawbacks. First, these systems often use large lexical-semantic databases that in addition to linguistic models contain a description of large semantic structures, e.g. ontologies. The databases are built mainly by hand and, as a rule, are specialized for a narrow subject area. Consequently, these systems are hard and expensive to scale for new subject areas due to the annotation cost.

For neural networks, as well as for structural methods, forming a training set is also a serious challenge, especially considering that after training or building a knowledge base there has to be enough material to form a coherent text on request.

Coherence Metrics. The solution of the task is complicated by the lack of a single accepted effective metric for assessing the quality of the generated texts. There are works in which BLEU [11] is used as such metric, while originally having been developed to assess the quality of machine translation. It is well able to identify paraphrase, plagiarism, and other similarities between texts [12]. It can be used to measure the relevance of the generated text to the request. However, the aim of this research is to create texts that are not identical to the request. Since BLEU will produce the highest values for a set of verbatim paraphrases of the input request, its use in tasks similar to the one at hand can be limited.

Taking into account our goal of creating coherent texts, we have to consider coherence models and metrics. Various approaches to control and measure coherence were proposed in a number of works.

Foltz, Kintsch, and Landauer's [13] model measures coherence as a function of semantic relatedness between adjacent sentences. Semantic relatedness is computed automatically using Latent Semantic Analysis from raw texts without employing syntactic or any other annotation. Barzilay and Lapata [14] propose an approach that captures local coherence by modeling patterns of entity distribution in discourse using an entity-grid model.

Recent deep learning coherence works [15, 16] adopt recursive and recurrent neural networks for computing semantic vectors for sentences. Coherence models that use recursive neural networks often depend severely on external resources, e.g. syntactic parsers, to construct their recursion structure. Coherence models that rely purely on recurrent neural networks process the words within a text sequentially. However, in such models, long-distance dependencies between words cannot be captured effectively due to the limits of the memorization capability of recurrent networks.

Basile et al. [17] describe a relatedness model built using FrameNet, which formally describes semantic structures of predicate-argument relations of English clauses. This allows the authors to consider the relatedness of two clauses as a sum of two estimates: the relatedness of the predicates of these two clauses and the relatedness of the corresponding arguments of these two predicates.

Mesgar and Strube [18] propose a neural model of local text coherency using recurrent networks and a convolutional network and demonstrate state-of-the-art results for two tasks: readability assessment and essay scoring.

METHOD

S is a set of sentences relevant to a query consisting of a set of keywords. A set of sentences S is formed as a result of a search in the corpus of texts. The search is performed using a search index built over the text corpus using the search platform SOLR (http://lucene.apache.org/solr/).

The search index is implemented as an inverted index. Each word from the corpus provided with a list with sublists:

$$w\colon ((d_1,((n_{11},s_{11})(n_{12},s_{12})\dots(n_{1k},s_{1k}))),\dots$$
$$\dots,(d_r((n_{r1},s_{r1})(n_{r2},s_{r2})\dots(n_{rt},s_{rt})))), \quad (1)$$

where d_i is the index of the document containing this word w; n_{ij} is the index of the word w in the text d_i; and s_{ij} is the index of the sentence containing the given word w in document d_i.

As a result of a search by request, sentences that contain at least two of the k words from the query are selected from the corpus of texts, provided that the paragraph of each of the returned sentences (or its entire text, if it is sufficiently short — three paragraphs or less) contains all query keywords or their synonyms. The next task is to assemble a coherent text of three sentences from the set S. We emphasize that all sentences must be taken from different texts.

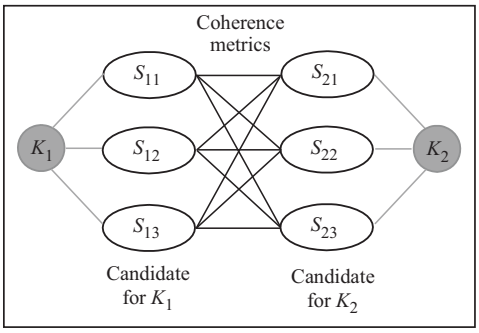

Fig. 1. Process of sentence combination to create a unique coherent text with respect to the proposed metrics (K_1 and K_2 are subsets of keywords from the query)

To combine sentences into text, a hierarchical agglomerative clustering algorithm is used. At the first stage of the algorithm, each sentence is a cluster. Then the nearest pair of clusters is selected and merged into one (see Fig. 1).

The metric for finding the nearest sentences is calculated in a similar way to [17] using the formula for calculating relatedness:

$$FRel(S_1,S_2)=\alpha\times FRelPred(S_1,S_2)+(1-\alpha)\times FRelArgs(S_1,S_2), \quad (2)$$

where *FRelPred* is the relatedness of predicates:

$$FRelPred\ (S_1,S_2)=\log_2\frac{|Cp_1p_2|}{|Cp_1||Cp_2|}, \quad (3)$$

where Cp_1 and Cp_2 are subsets sentences from the corpus that have common predicate p_1 with the first sentence S_1 and p_2 with the second sentence S_2, respectively. $Cp_1 p_2$ is a subset of contexts– the adjacent sentences in the text corpus, where p_1 and p_2 are the main verbs-predicates of the first and the second sentences respectively.

FRelPred is similar to the PMI (pointwise mutual information) metric. It can be calculated dynamically using a special inverted index, constructed after preprocessing the sentences in the text corpus with a part-of-speech tagger. The resulting special index contains only verbs that are predicates of the sentences in the corpus.

It should be mentioned that order matters in (3). Due to language nature, *FRelPred* is a quasimetric that does not satisfy the condition of symmetry. Changing the order of sentences in the text will likely change the meaning or affect its coherence.

Relatedness for predicate arguments is calculated as follows:

$$FRelArgs(S_1, S_2) =$$

$$= \frac{1}{2}\left(\frac{1}{|\arg s_1|} \sum_{\substack{N_i \in arg\, s_1 \\ N_j \in \arg s_2}} \max wpsim(N_i, N_j) + \frac{1}{|\arg\, s_2|} \sum_{\substack{N_i \in \arg s_2 \\ N_j \in \arg s_1}} \max wpsim(N_i, N_j) \right), \quad (4)$$

where $\arg s_1$ and $\arg s_2$ are sets of noun arguments of verb predicates p_1 and p_2, respectively, *wpsim* is the Wu-Palmer similarity measure for two words [19], α is the coefficient of balancing between the two components of (2). Basile suggests the optimal value $\alpha = 0.5$ obtained experimentally [17].

Wu and Palmer calculate relatedness of two concepts-words by considering depths of the two corresponding synsets in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer, or closest common ancestor):

$$wpsim(S_1, S_2) = \frac{2 \times depth(LCS(S_1, S_2))}{depth(S_1) + depth(S_2)}. \quad (5)$$

Fig. 2 demonstrates the process of sentence combination according to (2). Solid outline and double outline boxes identify nouns, connections between which are found using (4). Relations between verbs predicates depicted in dotted outline boxes calculated with (3).

First, we select and concatenate such sentences S_1 and S_2, forming two-sentence clusters, for which the following condition is satisfied:

$$FRel(S_1, S_2) \geq R, \quad (6)$$

where R is a threshold found empirically. Sentences can go into several separate clusters at once.

At the next stage of the algorithm, when singular sentences join clusters of two sentences, we try to put them either as the first or the last sentence constructed texts. In this case, condition (6) for the new sentence with the adjacent one should be satisfied.

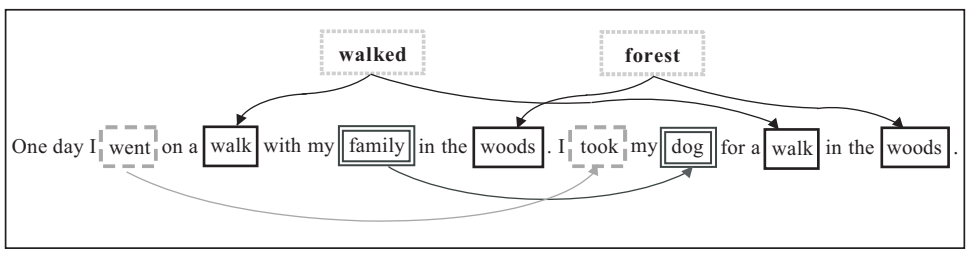

Fig. 2. The example of search

When two clusters sharing a sentence are formed — e.g., (S_1, S_2) and (S_2, S_3) — they are merged together to form cluster (S_1, S_2, S_3). The coherence metric of the constructed text is calculated by the formula:

$$Coherence(T) = \frac{\sum_{i=1}^{n-1} FRel(S_i, S_{i+1})}{n-1}. \quad (7)$$

We accept the text corresponding to the maximum value of $Coherence(T)$, provided that

$$Coherence(T) \geq P, \quad (8)$$

where P is a threshold. Its optimal value also figured out in the experiment when calculating $Coherence(T)$ for paragraphs from the original texts of the corpus. The algorithm can potentially continue the assembly of texts of any length, but for the current research, we have limited our scope to shorter texts of no more than three sentences.

The flaw of the method is the need to enumerate all possible variants of the sentences found by the search engine of the system from the corpus of texts. There are hundreds and even thousands of sentences that are relevant to a query, which makes a complete search through possible pairs and triples of sentences almost impossible when operating in real time. Randomly reducing the set of sentences can lead to a lack of pairs or triples of sentences satisfying the conditions of coherence (6) and (8). It is necessary to effectively filter the initial set of sentences found by the system's search engine, leaving a subset of ones most relevant to query. We propose to use the word2vec vector model to construct a vector metric for proximity of sentences to a given query. In the word2vec model, each word corresponds to a certain vector. The vector of a sentence will be calculated as a weighted sum of vectors of all meaningful words in this sentence. The words for which the TF-IDF metric exceeds a certain threshold level *Th* are considered meaningful. The weight of each vector in the sum is proportional to TF-IDF of its word. Query vector is calculated in a similar way. We use cosine as proximity metric. Having a query vector and vectors of N sentences selected from the text corpus, it is possible to select from the N sentences k of them that are most relevant to query for linear time. We can also reduce this search time to logarithmic time $O(\log N)$. For this purpose, we have to make a special pre-processing of the corpus.

We propose to implement hierarchical agglomerative clustering for all sentences in the text corpus.

Step 1. All sentences in the corpus are separate single clusters.

Step 2. While we don't have one global merged cluster do

— Select the nearest pairs of clusters according to the cosine measure;

— Merge the nearest pair of clusters;

— Recalculate centroids in new clusters.

After merging all the clusters into one global cluster, we will have a binary search tree of sets of sentences.

Having received the query vector, we go to the root of the search tree, and begin to compare the centroid vectors of the left and right son of the given node with the query vector by means of the cosine measure. After measurement we go in the direction of the best option. Having done this several times, we will find the desired subset of sentences Q that satisfy the criterion of proximity to query vector with a given threshold level or for the required number of sentences. After that, text assembly works only with found sentences that are included in subset Q. The text corpus together with the search index for the system is an implicit knowledge base that stores information about the realities of the surrounding world and subject areas described in the texts. The building of such a knowledge base is fully automatic, as is its

replenishment by including additional texts to the corpus and automatically updating the search index.

In Fig. 3, the general pipeline of text creation based on the proposed coherence metrics is shown.

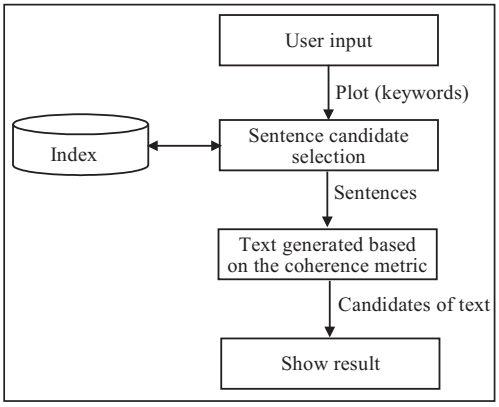

Fig. 3. The approach of text generation using the proposed coherence metric

EXPERIMENTS

We study the effectiveness of the metric by training a text generation model and evaluating it against two other approaches.

Dataset. We conduct the experiments on the ROCStories corpus [20]. It consists of 98,162 short commonsense stories containing on average 50 words each. Our system constructs a search index for this corpus and uses the index to form coherent texts. The index used for calculating (3) was built from a large collection of blog texts.

Baselines. To evaluate the effectiveness of the proposed approach, we compare our method against such representative baselines as the skip-thoughts model [6] specially trained on the above-mentioned corpus and the plan-and-write hierarchical generation system presented in [21] that actually also uses the ROCStories corpus.

Skip-thoughts is an encoder-decoder model. Decoder consists of two parts: one predicts the previous sentence for the current one, and another — the next one. As an encoder we use bidirectional RNN with GRU activations (64 units each). As decoders, we use two RNNs with GRU activations with 128 units. As an output, we get the vector of probabilities for all words from the dictionary. Dictionary size: 11 000 words. For word embeddings in encoder and decoder models, we use a pre-trained fasttext (https://fasttext.cc) model for simple English. The model was trained during 500 steps (16000 randomly chosen samples were given at each step).

Model Training. Training of the proposed model is conducted with calculating the optimal values for parameters R and P (conditions (6) and (8) correspondingly). We partition all text corpus into 3 equal parts. The first part is left as is. The second part has been mixed to form incoherent text set. In this case, in constructed texts, every sentence is selected from a distinct text. In the third part of the corpus, only half of the texts were mixed in the same way as mentioned above and they are marked as incoherent.

Several machine learning models were used to figure out the optimal values for R and P. The first and the second part of corpus were used as a training set and the third as a test set.

The best results of classification on test set were obtained by a support vector classifier with parameters $R = 52.81$ and $P = 59.16$.

After the training, the three systems receive a query for a future story as an input and generate their texts. Typical outputs of each system are shown in Table 1.

Evaluation. For the expert evaluation study, we offered 50 fluent English speakers to assess texts generated by the three approaches without disclosing which system generated which passage. Every expert received a questionnaire with texts generated according to 20 queries. The queries were taken from the pool of image captions in the COCO dataset [22], making sure that all the words in the chosen caption are present in the ROCStories dataset. For each text, three parameters were evaluated on a scale from 1 to 10: Readability (coherency and consistency), Likability (interestingness) and Appropriateness (relevance to the query). Averaged estimates are shown in Table 2.

Table 1. Examples of generated stories

Input query	A man riding skis down a snow covered slope
Skip-thoughts	1. He landed on the ground. He landed on the ground. He ran to the hospital. 2. He ran to the hospital. The doctor diagnosed him. The doctor diagnosed him.
Plan-and-write	1. It was a cold winter day. John decided to shovel the snow. He put on his boots and went outside. He decided to go for a walk. He was covered in snow. 2. Tim was driving down the road. He saw a car in the driveway. He whipped it up and took it home. He put it in his truck. He decided to keep it.
Proposed solution	1. This morning,I woke up and saw the ground in snow. The streets were covered in snow. 2. He fell down a snowy hill. When he got up he was covered in snow. He saw that his walkway was covered with deep snow.
Input query	Several large commercial airplanes stationed at an airport with service vehicles nearby.
Skip-thoughts	1. In the end, the flight was delayed. The flight was delayed. The plane was delayed. 2. The plane was delayed. The flight was delayed. The plane was delayed.
Plan-and-write	1. I was on my way to the airport. I got my luggage. I was waiting in line. I pulled over. I realized I missed the flight. 2. Tom was on his way to the airport. He was on his way to his flight. He couldn't find his windows. He called the cops. They had to call the police.
Proposed solution	1. A large family went to the airport. It was their first time flying a commercial plane of a large airline. 2. The plane shook as it rose into the air. The weather was changing when the plane was high in the air.

Table 2. Results of human estimation

System	Readability	Likability	Appropriateness	Overall
Skip-thoughts	3.714	3.281	3.433	3.476
Plan-and-write	5.161	4.212	5.369	4.914
Our system	6.751	6.281	6.004	6.345

For Readability assessment, the coefficient of variation of human expert estimations on the given set of generated texts was between 6% and 14% (as shown in Table 3) that demonstrated acceptable quality of coherency assessment. For Appropriateness assessment, the coefficient of variation was between 5% and 11%. For Likability assessment, the coefficient of variation was between 16% and 23% due to the extra subjectivity of that specific part of estimating.

We propose to deploy (2) as the metrics of coherency for obtained texts. We can calculate its values for all three sets of texts but there is a problem due to the essence of our method. It selects such combinations of sentences that already satisfy the found optimal values of parameters R and P. Still, we present the results of such measuring in Table 4. During the experiments, it was found that the Spearman correlation between the values of $Coherence(T)$ and the human assesses of $Readability(T)$ was 0.86, which indicates the quality of the proposed metrics and the effectiveness of the method as a whole.

Results and Discussion. As can be seen from Table 2, the proposed method overcomes competitors in all assessments. The results obtained during the expert assessment may be hard to reproduce since the evaluation fully relies on human judgment, but unfortunately for this specific case there are no better way to asses such subjective qualities of text as coherency, appropriateness and especially likability without participation of people.

Table 3. Coefficient of variation (CV) in %, calculated for expert assessments of Readability of stories generated for 20 queries (Q) by three systems: encoder-decoder (Sl), plan-and-write (S2), and our system (S3)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
S1	7.1	9.8	10.2	8.8	8.6	12.2	7.9	13.7	11.9	12.7	13.5	11.9	12.4	10.5	13.6	12.8	11.1	12.3	11.1	10.8
S2	9.8	13.8	10.7	11.2	12.2	10.3	9.1	12.9	8.3	10.1	12.4	13.2	9.9	13.9	8.4	8.9	7.3	8.8	11.9	9.5
S3	11.0	8.5	9.4	12.9	11.7	9.8	11.3	13.1	10.4	13.2	8.6	11.7	8.7	10.9	11.7	11.8	6.4	9.8	9.1	10.0

Table 4. Estimation of introduced metrics

System	*FRel*	*Coherence*
Skip-thoughts	27.12	34.82
Plan-and-write	39.65	45.91
Our system	60.18	64.15

There are no widely available AI systems that read and understand natural language texts better than humans. Comparison with other existing approaches to measuring the coherency of texts is an interesting but technically difficult to implement experiment, and discrepancies between different metrics still lead to the need for human experts to choose a more correct option. In our experiment, good coefficient of variation of human expert estimations ensures the reliability of obtained estimation. For the second part of the experiment, one can note that it is not quite correct to evaluate baselines and the proposed method on *FRel* and *Coherence*, since the two metrics are locally defined in this work, and the proposed method is specifically designed to optimize on them while the compared baselines are not. *FRel* and *Coherence* for baselines are shown not to demonstrate the advantage of the approach, but to measure the correlation of calculated values of our metrics and estimations obtained during expert assessment. And the obtained high correlation demonstrates good quality of the proposed metrics.

One of the flaws of the metrics is its excessively high evaluation of texts with repeated almost or completely identical sentences generated by neural networks. This influence of repetitions on the metrics can be compensated by imposing a penalty for repeats, which are successfully identified by the BLEU metrics: If the text contains sentences S_i and S_j, such that $BLEU(S_i, S_j) > 0.8$, then $Coherence(T) = Coherence(T) - Penalty$.

Additionally, it should be mentioned that the proposed approach is applicable for usage in standalone on-device running for mobile devices. The average text generation time per one request is ~ 2.30 sec, which is comparable with the analyzed NN-based solutions. But the size of the necessary components is much lower: for our solution the ROCStories corpus size is 27 MB and the SOLR index size is ~ 500 MB, while in case of NN-based generation the combined size of required components is about 2.7 GB.

CONCLUSIONS

This paper proposes the use of coherence metrics during text generation. We demonstrate the advantages of controlling coherence of the output by implementing a system for assembling coherent texts based on a search engine with special search indexes that uses the proposed coherence metrics and conducting an expert study.

The search system selects from the text corpus a set of sentences that relevant to the request, after that the system composed a coherent story from them. Coherence metrics that serve as the main criterion for combining sentences into text can be calculated dynamically using data from the built indices. The experimental results confirm the reliability of the proposed metrics and the effectiveness of the method as a whole. Compared to NN-based approaches, the proposed method is more applicable for use on mobile devices due to lower storage and computation power requirements.

Future work includes conducting a more extensive expert study involving comparison of auto-generated texts with human-written ones; studying the behavior of the proposed metrics in longer texts; and integrating text coherence control into NN-based approaches.

REFERENCES

1. Ruder S., Ghaffari P., Breslin J.G. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. 2016. URL: http://arxiv.org/abs/1609.06686.
2. Agarwal B., Ramampiaro H., Langseth H., Ruocco M. A deep network model for paraphrase detection in short text messages. *Information Processing and Management.* 2018. Vol. 54, N 6. P. 922–937.
3. Asghar N., Poupart P., Hoey J., Jiang X., Mou L. Affective neural response generation. Advances in Information Retrival. *Proc. 40th European Conf. on IR Research, ECIR 2018* (March 26–29, 2018, Grenoble, France). P. 154–166.
4. Chen Y.-N., Celikyilmaz A., Hakkani-Tur D. Deep learning for dialogue systems. *Proc. 27th Int. Conf. on Computational Linguistics: Tutorial Abstracts.* 2018. P. 25–31. URL: https://www.aclweb.org/anthology/C18-3006.pdf.
5. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog. URL: https://openai.com/blog/better-language-models/ (Feb 14, 2019).
6. Kiros R., Zhu Y., Salakhutdinov R.R., Zemel R., Urtasun R., Torralba A., Fidler S. Skip-thought vectors. *Proc. NIPS 2015.* (December 7–12, 2015, Montreal, Quebec, Canada). Vol. 2. P. 3294–3302.
7. Jain P., Agrawal P., Mishra A., Sukhwani M., Laha A., Sankaranarayanan K. Story generation from sequence of independent short descriptions. 2017. URL: http:// arxiv.org/abs/1707.05501.
8. McIntyre N., Lapata M. Learning to tell tales: a data-driven approach to story generation. *Proc. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP* (2–7 August 2009, Suntec, Singapore). 2009. P. 217–225.
9. Rishes E., Lukib S.M., Elson D.K., Walker M.A. Generating different story tellings from semantic representations of narrative. *Proc. 6th ICIDS 2013*. (November 6–9, Istanbul, Turkey). P. 192–204.
10. Leppänen L., Munezero M., Granroth-Wilding M., Toivonen H. Data-driven news generation for automated journalism. *Proc. 10th INLG 2017*. (September, 2017, Santiago de Compostela, Spain). P. 188–197.
11. Papineni K., Roukos S., Ward T., Zhu W.-J. BLEU: A method for automatic evaluation of machine translation. *Proc. 40th Annual Meeting on Association for Computational Linguistics*. (July, 2002, Philadelphia, Pennsylvania, USA). 2002. P. 311–318.
12. Ji Y., Eisenstein J. Discriminative improvements to distributional sentence similarity. *Proc. 2013 Conference on Empirical Methods in Natural Language Processing.* (October, 2013, Seattle, Washington, USA). P. 891–896.
13. Foltz P. W., Kintsch W., Landauer T.K. The measurement of textual coherence with latent semantic analysis. *Discourse Processes.* 1998. Vol. 25, N 2–3. P. 285–307.
14. Barzilay R., Lapata M. Modeling local coherence: an entity-based approach. *Computational Linguistics.* 2008. Vol. 34, N 1. P. 1–34.
15. Li J., Hovy E. A model of coherence based on distributed sentence representation. *Proc. EMNLP 2014.* (October 25–29, 2014, Doha, Qatar). P. 2039–2048.
16. Li J., Jurafsky D. Neural net models of open-domain discourse coherence. *Proc. EMNLP 2017*. (September, 2017, Copenhagen, Denmark). P. 198–209.
17. Basile V., Condori R.L., Cabrio E. Measuring frame instance relatedness. *Proc. 7th Joint Conference on Lexical and Computational Semantics (*SEM)* (June 5–6, 2018, New Orleans). 2018. P. 245–254.
18. Mesgar M., Strube M. A neural local coherence model for text quality assessment. *Proc. 2018 Conference on Empirical Methods in Natural Language Processing* (October 31 – November 4, 2018, Brussels, Belgium). P. 4328–4339.
19. Wu Z., Palmer M. Verbs semantics and lexical selection. *Proc. 32nd Annual Meeting on Association for Computational Linguistics* (June 27–30, 1994, Las Cruces, New Mexico). 1994. P. 133–138.
20. Mostafazadeh N., Chambers N., He X., Parikh D., Batra D., Vanderwende L., Kohli P., Allen J. A corpus and cloze evaluation for deeper understanding of commonsense stories. *Proc. NAACL-HLT 2016.* (June 12–17, 2016, San Diego, California). P. 839–849.
21. Yao L., Peng N., Weisahedel R., Kbight K., Zhao D., Yan R. Plan-And-Write: Towards better automatic storytelling. 2018. URL: https://arxiv.org/abs/1811.05701.
22. Lin T.-Y, Maire M., Belongie S., Hays J., Peroba P., Ramanan D., Dollar P., Zitnicl L. Microsoft COCO: Common objects in context. *Proc. ECCV 2014.* (September 6–12, 2014, Zurich, Switzerland). P. 740–75.

Надійшла до редакції 15.08.2019

О.О. Марченко, О.С. Радивоненко, Т.С. Ігнатова,
П.В. Титарчук, Д.В. Железняков

ПОКРАЩЕННЯ ЯКОСТІ ГЕНЕРУВАННЯ ТЕКСТУ ЗА ДОПОМОГОЮ МІРИ ЗВ'ЯЗНОСТІ

Анотація. Взаємодія, що ґрунтується на тексті з використанням мобільних пристроїв, стала повсюдною, її основними джерелами є соціальні мережі, месенджери, електронні листи, віртуальні помічники, застосунки для забезпечення доступності тощо. Це передбачає потребу у створенні систем полегшення введення тексту користувачем та розробленні способів підтримки вербального зворотного зв'язку. Описано метод генерації унікального тексту для мобільних пристроїв та методологію його оцінювання як розв'язки обох зазначених вище задач. Розглянуто можливості, надані використанням контексту (місцезнаходження, погода, заплановані події тощо), обмеження обчислювальних ресурсів та використання даних, а також притаманну суб'єктивність оцінювання творчої задачі з урахуванням різноманіття можливих прийнятних результатів. Порівняння з іншими методами генерації текстів свідчить про те, що використання метрик зв'язності дає змогу досягти більш високого рівня якості з погляду сприйняття людиною. Кореляція Спірмена між значеннями пропонованої метрики та оцінкою читабельності тексту людиною становить 0.86, що свідчить про високу якість метрики та ефективність методу в цілому.

Ключові слова: комп'ютерна лінгвістика, автоматичне генерування природномовних текстів, зв'язність текстів, метрики зв'язності текстів.

А.А. Марченко, О.С. Радивоненко, Т.С. Игнатова,
П.В. Титарчук, Д.В. Железняков

УЛУЧШЕНИЕ КАЧЕСТВА ГЕНЕРАЦИИ ТЕКСТА С ПОМОЩЬЮ МЕРЫ СВЯЗНОСТИ

Аннотация. Взаимодействие на основе текста с использованием мобильных устройств стало повсеместным, его основными источниками являются социальные сети, мессенджеры, электронные письма, виртуальные помощники, приложения для обеспечения доступности и т.д. Это подразумевает необходимость создания систем облегчения ввода текста пользователем и разработки способов поддержки вербальной обратной связи. В этой статье мы обсуждаем метод генерации уникального текста для мобильных устройств и методологию его оценки в качестве решения обеих заявленных проблем. Мы рассматриваем возможности, предоставляемые использованием контекста (местоположение, погода, запланированные события и т.д.), ограничения вычислительных ресурсов и использования данных, а также присущую субъективность оценки творческой задачи с учетом разнообразия возможных приемлемых результатов. Сравнение с другими методами генерации текстов показывает, что использование метрик связности помогает достичь более высокого качества с точки зрения человеческого восприятия. Корреляция Спирмена между значениями предлагаемой метрики связности и человеческой оценкой читабельности текста составляет 0.86, что свидетельствует о высоком качестве метрики и эффективности метода в целом.

Ключевые слова: компьютерная лингвистика, автоматическая генерация естественно-языковых текстов, связность текстов, метрики связности текстов.

Marchenko Oleksandr,
Dr. Sc. (Phys.-Math.), Professor, Taras Shevchenko National University of Kyiv, Ukraine, e-mail: rozenkrans17@gmail.com.

Radyvonenko Olga,
Ph.D in Technical Science, Associate Professor, Head of Lab, Samsung R&D Institute Ukraine (SRK), Kyiv, Ukraine, e-mail: o.radyvonenk@samsung.com.

Ignatova Tetiana,
Engineer, Samsung R&D Institute Ukraine (SRK), Kyiv, Ukraine, e-mail: te.ignatova@samsung.com.

Tytarchuk Pavlo,
Engineer, Samsung R&D Institute Ukraine (SRK), Kyiv, Ukraine, e-mail: p.tytarchuk@samsung.com.

Zhelezniakov Dmytro,
Staff Engineer, Samsung R&D Institute Ukraine (SRK), Kyiv, Ukraine, e-mail: d.zheleznyak@samsung.com.